Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

A Reconfigurable Neural Signal Processor (NSP) for Brain Machine Interfaces

Shalom Darmanjian$^{1}_{a}$, Grzegorz Cieslewski$^{1}_{a}$, Scott Morrison^{2}, Benjamin Dang^{3}, Karl Gugel$^{1}_{b}$, Jose Principe$^{1,2}_{c}$
IEEE Student Member$_{a}$, IEEE Member$_{b}$, IEEE Fellow$_{c}$

***Abstract*— In this paper, we present a design for a wearable computational DSP system that alleviates the issues of a previous design and provides a much smaller and lower power solution for the overall BMI goals. The system first acquires the neural data through a high speed data bus in order to train and evaluate prediction models. Then it wirelessly transmits the predicted results to a simulated robot arm. This system has been built and successfully tested with real and simulated data.**

I. INTRODUCTION

The ultimate goal of brain machine interfaces (BMIs) is to offer paralyzed patients a communication path between their brain and the external world without inhibiting their mobility. Current BMI technology hinders the patient's mobility since it involves large machines and multiple wire connections. These systems first start with analog neural signals being recorded from a patient's motor cortex. Then these low-voltage signals are passed through amplifiers and spike detection hardware (and software) in large rack-mountable DSP processor boards [1], [15], [6]. After this pre-processing step, binned neural data is then sent to Matlab-enabled PCs for real-time prediction of the patient's arm movement as they engage in a task [1], [15], [6]. Finally, the trajectory data is sent to a robotic arm to reproduce the predicted trajectory of the patient's arm (see Figure 1).

Unfortunately, this tethered cluster of machines falls short from the ultimate goal of having patients interacting in the world without physical inhibition. Some researchers are trying to remove this tether by wirelessly transmitting the analog neural signals off the patient. In fact, a majority of the BMI research solely focuses on shrinking the wireless acquisition hardware [8], [7], [14], [9].

Although wirelessly transmitting the analog neural signals is more inline with the ultimate BMI goal, this approach is also flawed since it still requires large immobile machines to predict trajectories [8], [9]. Additionally, there are bandwidth limitations with this solution. For example, if more neurons are sampled from the brain, the transmission of these signals requires higher powered transceivers. These higher powered transceivers then reduce the experimental time of the system or require larger batteries that reduce mobility. Since the current push for BMIs is to sample from as many neurons as possible, there is a need to address these current and future bandwidth issues [1].

The next evolutionary step in BMI hardware should solve these issues by shrinking the digital hardware so that the prediction algorithms are computed on the patient, rather than off-board [11]. Our proposed solution is to directly connect the analog and digital subsystems with a high-speed data bus that is more power efficient and faster than any wireless link.

[1]Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611-6200
[2]Department of Biomedical Engineering, University of Florida, Gainesville, FL 32611-6200
[3]Texas Instruments,12500 TI Blvd, M/S 8731, Dallas, Texas, 75243, USA
[4]Lockheed Martin IS&S 3120 Zanker Rd. San Jose, CA 95134 USA

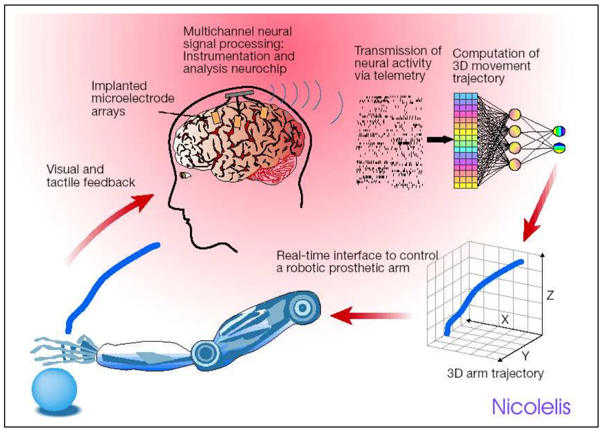
Multichannel neural signal processing: Instrumentation and analysis neurochip
Implanted microelectrode arrays
Transmission of neural activity via telemetry
Computation of 3D movement trajectory
Visual and tactile feedback
Real-time interface to control a robotic prosthetic arm
Z
X
Y
3D arm trajectory
Nicolelis

Fig. 1. Project Overview

Our previous work dealt with developing a digital system to solve these mobility and bandwidth issues and included an 802.11B PCMCIA card and DSP processor [11]. Unfortunately, the power consumption and size of the 802.11B DSP system was too large for a mobile BMI system [11]. Additionally, since the system was built for a single application, it is too rigid to address the changing needs of BMI systems.

In this paper, we present a design for a wearable computational DSP system that alleviates the shortcomings of the previous design. It provides a much smaller and lower power solution for the overall BMI goals. The system first acquires the neural data through a high speed data bus in order to train and evaluate prediction models. Then it wirelessly transmits the predicted results to a simulated robot arm. This system has been built and successfully tested with real and simulated data. We also tested this system with a simple spike detection algorithm on simulated data.

The organization of the paper is as follows. We first outline the system design in terms of the hardware modules and the software layers. Then we present the performance of the system in the results section, followed by a discussion and conclusion section.

II. SYSTEM DESIGN

A. *Overview of System Requirements*

The system described in this paper serves as the digital module of the overall BMI structure and is responsible for mapping the neural firings into action in the external world. To do this mapping, the system first acquires digitized neural data from the analog module. After acquisition and spike detection, the system then computes a linear or non-linear model to map the neural data to a trajectory. Once a trajectory is determined, the system finally transmits commands wirelessly to an off-board robotic arm.

Since the system needs to be carried by a patient (human, primate, or rat) without wires, there are portability and power

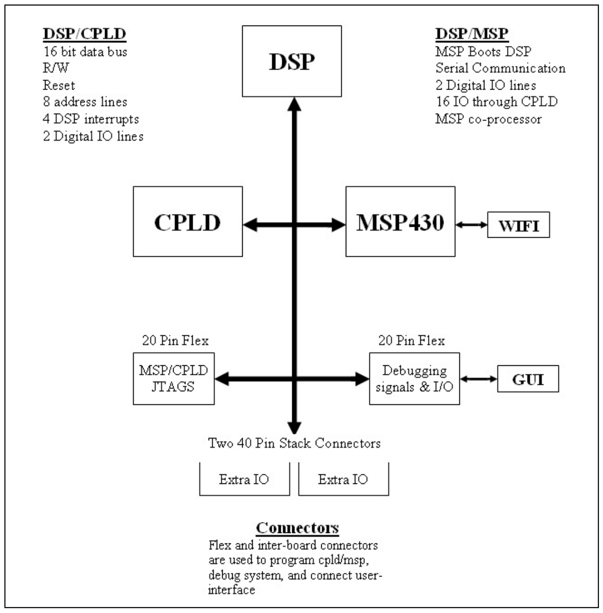

Fig. 2. Modules of BMI system

constraints. In turn, these constraints will affect the design choices. First, the size and weight of the PCB board and its components must be small enough to accommodate various patients. Second, the power consumption must be low in order to use lighter batteries and yield longer experimental trials.

With these power and size constraints placed on the system, the choice of processor and wireless transceiver becomes a concern. Adding to the challenge, the prediction models require decimal computations; therefore the processor needs to support floating point numbers. It must also have enough speed to attain real-time model computations, yet low power enough to meet the system constraints. Likewise, the wireless transceiver must not only be small with a sufficient transmission bandwidth for current and near-future needs, but it must also be low power. In the following subsections, we detail the NSP component choices that meet these system requirements.

B. DSP

The central component to any computational system is the processor. The processor has the ability to determine the speed, computational throughput and power consumption of the entire system. Moreover, it dictates what additional support devices will be used for the overall system design. For the NSP, we chose the Texas Instruments TMS320VC33 (C33) Digital Signal Processor (DSP). We chose this processor due to its suitability for on-board BMI modeling [2], [10]. It satisfies the computational needs and power consumption while still retaining versatility for future needs.

The C33 meets high speed and floating-point requirements since it is a floating-point DSP capable of up to 200 MFLOPS (with over clocking). It achieves such high speeds by taking advantage of its dedicated floating point/integer multiplier. Additionally, it works in tandem with the ALU, allowing it to compute two mathematical operations in a single cycle. With support for both assembler and C/C++, code for the DSP can be optimized at low and high programming levels for additional speed.

Even at 200MFLOPS, the C33 is also well suited to low power needs since it uses less than 200mW. It achieves such power savings due in part to its 1.8V core and other power saving measures built into the processor [4]. Using adjustable clock speeds, the DSP can exchange processing speed for lower power consumption. This flexibility is desirable since some BMI experiments may require increased battery life in exchange for less computational speed (i.e. less neural data).

The C33 is able to fulfill expandability requirements of the system as well. First, it supports a large address space by providing a 24-bit address bus to read and write 16 million different locations. This address space is used to map the different hardware components of the system and future hardware interfaces. It includes a 32bit parallel data bus and a serial port that allows for multiple communication methods and hardware multiplexing. This processor can also provide expansion since it has four hardware interrupts, two 32-bit timers, and a DMA controller.

C. Wireless Communication

The wireless connection is the second most important hardware component in the NSP. This component, similar to the processor, has the ability to define the size and power consumption of the full system. With our previous DSP design, we determined that 802.11B was the most appropriate protocol for our group and collaborators to interface with [11], [14]. The protocol not only provides a large amount of bandwidth, but also has a large code infrastructure for communication clients and servers. Unfortunately, it required a PCMCIA card that was very large and consumed the majority of the system's power [11].

Since the BMI modeling is computed on board, the NSP does not need a lot of bandwidth to transmit 100ms trajectory coordinates (20 bytes). Therefore, we can trade the bandwidth of the PCMCIA card for a 200X smaller wireless solution. This wireless chip, Nordic nRF2401, is a single 2.4Ghz chip transceiver capable of 1Mbps at a cost of 10.5mA (peak). It also has the ability to reduce its power further to 0.8mA when placed into ShockBurst mode. Compared to the 100mA-300mA PCMCIA card (depending on throughput), this wireless chip provides a significant savings in power. Additionally, the package size of the Nordic chip is only 5mm x 5mm compared to the 114.3mm x 57.2mm PCMCIA card used in the previous design. Although there is a separate antenna, it is only a 6.5mm x 2.2 mm surface mount SMD.

D. MSP Co-Processor/Boot-Loader

The NSP contains a co-processor to alleviate the computational strain on the DSP. This co-processor handles the peripheral functions (and any future functions) of the system, allowing the DSP to allocate resources for the algorithms that compute the neural-to-trajectory mappings. We chose the MSP430F1611x to serve as the co-processor since it is a very low power, yet highly functional, microcontroller.

This 16-bit RISC microcontroller consumes 1μA of power at 1MHz and only linearly increases to its maximum 15mA at 8MHz. Additionally, there are other power modes that allow it to use smaller amounts of power with voltage ranges from 1.8V to 3.6V [12]. For the NSP, we use 3.3V to supply the microcontroller. Further, the microcontroller has 48KB of flash ram that can retain data when power is off and additional 10KB of volatile RAM when powered. Combined with a 12bit A/D-D/A, and multiple asynchronous/ synchronous serial ports, and SPI/I2C ports, this microcontroller is very well suited to the task (10mm x 10mm).

Handling the wireless protocol is one of the main functions for this co-processor. It provides transparency to the DSP when it needs to send or receive wireless data. Underneath this transparency, the

MSP is handling the details of protocol control for the wireless chip.

We also use the MSP as a boot loader for the DSP by using the DSP's serial port for booting. Since the MSP has 48KB of flash, we store the DSP program in the flash before booting the DSP serially. Essentially, by using the MSP to boot the DSP, we trade one single-functioned component (EEProm) for a multifunctional component (MSP) that is lower power.

In terms of expandability, the MSP (8MIPS) is fast enough to handle future preprocessing of the neural data for the prediction models running on the DSP. Through a serial port or a digital IO (connected to the flex and interconnects), we can also expand the system to include USB communications for the uploading or downloading of data in real time (from multiple paths MSP/DSP/analog-module). We can also use these ports to create a test bed for debugging the system or testing future analog modules.

E. Reconfigurability

1) Complex Programmable Logic Device (CPLD): Using an Altera's EPM3128ATC100, the NSP is able to expand and accommodate future needs since it has access to all the connectors and components. There are options for using on-board/off-board clocks and even clocks from the analog module.

Since communication between the on-board components and off-board modules is critical, it is essential that the communication paths be reconfigurable for future needs. This includes the serial ports from the DSP and MSP, and the parallel data bus from both processors. Through the CPLD we have the ability to add a USB connection and other serial ports. In turn these can be redirected to the DSP or the MSP. The CPLD even has the ability to redirect the wireless connection directly to the external IO for debugging and support. Essentially, the CPLD is very useful for debugging the system and adding future features that may become necessary.

2) Connector Interface: In order to compliment the CPLD and allow for functionality of unknown future needs, it is necessary to provide the maximum amount of connections between the internal components on the NSP with the external modules. With this connectivity, we need connecters that have many contacts and are small enough to not increase the size of the system.

The first connection type must be able to mate with the analog modules and provide an adequate number of connections and yet be strong and compact enough to support the attached PCBs. For the NSP, we use two Hirose interconnects that allow for 80 different connections between the boards (see Figure 3). With this amount of connections there is support for the current needs of the analog module as well as support for daughter boards that may be designed in the future.

The second connection type is used for debugging and programming the DSP, CPLD, and MSP. In order to retain the ability to communicate with systems that may be at great distances, we chose a flex connecter interface. Specifically, two Hirose flex connectors provide 40 connections (Figure 3). Through these connectors there is access to the CPLD jtag, MSP serial ports, wireless chip, and other general CPLD IO for multiplexing communication between the components of the NSP.

3) Power Subsystem: The two different voltage requirements for the NSP are 1.8V and 3.3V. These voltages are not only used by the DSP, they are also used for the other components on the system [13]. In trying to supply the NSP components, the voltages can be supplied from multiple pathways. This includes both an on-board option and off-board option.

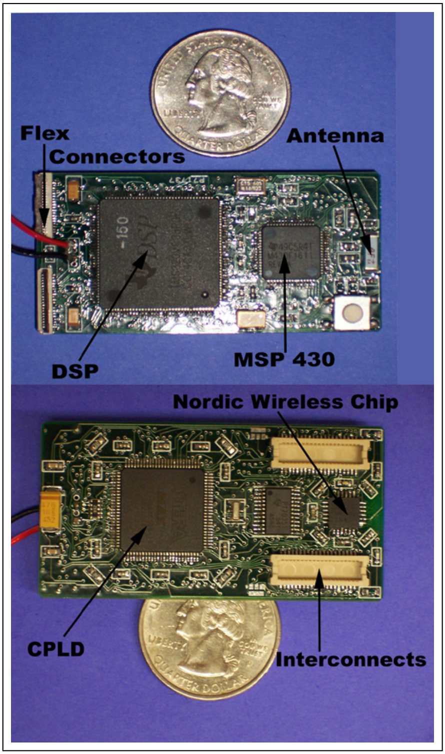

Fig. 3. NSP

We chose Texas Instruments TPS70351 Dual-Output LDO Voltage Regulator that provides both 1.8V and 3.3V voltages on a single chip and only requires 5V to operate as the on-board power option. This chip also provides the power-up sequence required by the DSP once it is initialized. Having a single supply is very desirable for a portable system since only one 5V battery supply is necessary.

With the off-board option, we can power the regulator through the flex connectors, interconnects, or simple headers, allowing a battery or external supply to power the system. Additionally, we can have the analog module or debugging system provide the power. This allows for the isolation of certain components when testing or debugging the system. With this flexability, future hardware can take advantage of the different powering options.

F. System Software

There are six layers of software in the NSP system environment: 1) PC Software, 2) DSP Operating System (OS), 3) DSP Algorithms, 4) VHDL Code 5) MSP430 Code 6) Client/Server code.

We wrote a PC console program to interface with the DSP through the USB. The console program calls functions within the DSP OS to initiate and control the USB communication functions. The DSP OS is also responsible for reading/writing memory locations and various program control functions. This is the main pathway for debugging the system.

In tandem with the OS layer of the DSP, there is low-level MSP code for initializing and controlling the wireless controller. This code must interact with the DSP OS and any client code or algorithms that are running simultaneously within the NSP. Once the prediction model completes an epoch or computation cycle, the program must interrupt the MSP and transfer any required data. This process also involves creating the correct packets for transmission to the appropriate server (off-board).

The final layer of code resides in the on-board Complex Programmable Logic Device (CPLD). This hardware-based VHDL code is responsible for correctly shuttling data between all of the hardware modules. It achieves this processing through a series of interrupts and control lines that are provided by the individual hardware components. Depending on the functionality of the system, this code is responsible for all the internal and external communication of the NSP system.

III. RESULTS

The NSP schematic capture was accomplished with Protel DXP. We sent the PCB board for production and populated it in our lab. The performance of each component section was evaluated before testing the full system (see Figure 3).

We used a PC to simulate the output of an analog module. We transmitted 104 digitized neural channels and 3-D trajectory data through a high-speed USB connection to the NSP. On the NSP, we computed the neural-trajectory mapping with a 10-tap FIR Filter (training was done with normalized LMS) [11], [2]. The average amount of time for a single prediction is 211 usec. Since we use the same DSP and LMS code, this test achieves the same timing results for the predicted output [11], [10], [2]. On a 600 Mhz PIII laptop running Matlab 6 the average prediction takes 23 msec. As explained in [11], [2], the factor of improvement or speed gain is around 100x for the DSP over the laptop. The LMS output results collected at the receiving computer were directly compared to Matlab computed outputs. These results are accurate within 7 decimal places of the Matlab double precision results.

The NSP wirelessly transmitted the 100ms trajectory coordinates to the receiver board. We sent repeated packets to compensate for potential packet loss. This was possible since the Nordic Nrf2401 can transmit at 1Mbps and we only needed 1kbps.

Additionally, the MSP successfully booted the DSP and transferred the BMI algorithm for computation. This involved the DSP and MSP serial ports for bi-directional communication. In a separate experiment, we were also able to test the parallel data bus for asynchronous communication between the systems. This experiment included the implementation of a simple spike detector (using loop thresholds) and a binning algorithm in the NSP (using simulated spikes).

Over the multiple experiments, the NSP consumed approximately 450-600mW with the on-board currents ranging between 90-120mA. In comparison, this is much less than the 1.75W and 4W previously attained by other acquisition hardware [8], [14], [7]. Additionally, the NSP board size is 63.5mmX 25.4mm and weighs 9grams.

IV. DISCUSSION AND CONCLUSION

We presented a design for a wearable computational DSP system that alleviates the shortcomings of our previous efforts. We demonstrated that the NSP could acquire the neural data through a high speed data bus in order to train and evaluate prediction models. It was also able to wirelessly transmit the predicted results to a simulated robot arm. Based on the smaller size, lower power consumption, and increased flexibility, this NSP should be a useful tool for future BMI experiments.

ACKNOWLEDGMENT

The authors would like to thank Dr. Nicolelis at Duke University. This work is supported by DARPA grant # N-66001-02-C-8022.

REFERENCES

[1] J. Wessberg, C. R. Stambaugh, J. D. Kralik, P. D. Beck, M. Laubach, J. K. Chapin, J. Kim, S. J. Biggs, M. A. Srinivasan, and M. Nicolelis et al., *Real-time prediction of hand trajectory by ensembles of cortical neurons in primates*, Nature, Volume 408, pp. 361-365, 2000.

[2] S. Morrison, *A DSP-Based Computational Engine for a Brain-Machine Interface*, M.S. Thesis, University of Florida, 2003.

[3] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue, *Brain-machine interface: Instant neural control of a movement signal*, Nature, vol. 416, pp. 141-142, 2002.

[4] Texas Instruments Incorporated, *TMS320VC33 Digital Signal Processor*, Literature Number SPRS087D, July 2002.

[5] J. Chapin, K. Moxon, *Neural prostheses for restoration of sensory and motor function*, Boca Raton (FL): CRC Press; 2000.

[6] J. K. Chapin, K. A. Moxon, R. S. Markowitz, M. A. Nicolelis, "Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex," Nature Neuroscience, vol. 2, pp. 664-670, 1999.

[7] Shahin Farshchi1, Istvan Mody2, and Jack W. Judy1,"A TinyOS-Based Wireless Neural Interface", IEEE EMBS San Francisco, CA, USA , September 1-5, 2004

[8] I. Obeid, M. Nicolelis, P. Wolf ., *A low power multichannel analog front end for portable neural signal recordings.*, J Neurosci Methods 2004;133:27-32.

[9] P. Irazoqui-Pastor, I. Mody and J. W. Judy, "In-vivo EEG recording using a wireless implantable neural transceiver" IEEE-EMBC Conf., March 2003.

[10] S. Morrison, J. Parks, K. Gugel, *A High-Performance Multi-Purpose DSP Architecture for Signal Processing Research*,Intl. Conf. on Acoustics, Speech, and Signal Processing, 2003.

[11] Shalom Darmanjian, Scott Morrison, Benjamin Dang, Karl Gugel, Jose Principe, "A Portable Wireless DSP System for a Brain Machine Interface," IEEE EMBS Neural Engineering Conference, 2005.

[12] Texas Instruments Incorporated, MSP430x1xx Family User's Guide (Rev. F) Texas Instruments,Literature number slau049f. Dallas (TX) USA; 2006.

[13] Texas Instruments Incorporated, *TPS70351 dual-output low-dropout voltage regulator.*, Literature number SLVS285. Dallas (TX) USA; 2000.

[14] I. Obeid, M. A. L. Nicolelis, and P. D. Wolf, "A multichannel telemetry system for single unit neural recordings," J. Neurosci. Meth., vol. 133, pp. 33-38, 2004.

[15] A. B. Schwartz, D. M. Taylor, and S. I. H. Tillery, "Extraction algorithms for cortical control of arm prosthetics," Current Opinion in Neurobiology, vol. 11, pp. 701-708, 2001.