A Portable Wireless DSP System for a Brain Machine Interface

Shalom Darmanjian^{1,2} Scott Morrison³ Benjamin Dang⁴ Karl Gugel² Jose Principe¹
oxygen@ufl.edu scott@ti.com benjamin.h.dang@lmco.com gugel@ecel.ufl.edu principe@cnel.ufl.edu

¹Computational NeuroEngineering Laboratory

²Applied Digital Design Laboratory

Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611-6200

³Texas Instruments, 12500 TI Blvd, M/S 8731, Dallas, Texas, 75243, USA

⁴Lockheed Martin IS&S 3120 Zanker Rd. San Jose, CA 95134 USA

Abstract—In this paper, we present a design for a wearable DSP system that is capable of processing various neural-to-motor translation algorithms. The system first acquires the neural data through a high speed data bus in order to train and evaluate our prediction models. Then via a widely used protocol, the low-bandwidth output trajectory is wirelessly transmitted to a simulated robot arm. This system has been built and successfully tested with real data.

I. INTRODUCTION

The feasibility of building brain machine interfaces (BMI) has been demonstrated with the use of digital computational hardware [1,3,4]. For these interfaces, researchers first acquire analog neural recordings and process them through spike detection hardware and software. Once the neural signals are processed, large rack-mountable high-end processors are used in conjunction with Matlab-enabled PCs to predict a subject's arm trajectory in real-time [1,4].

The ultimate BMI goal envisions that free-roaming subjects will possess the prediction algorithms and hardware in vivo as they physically interact in the world. Unfortunately, at the current stage of development, most researchers require the subjects to be tethered to a cluster of immobile machines. Other researchers have removed this tether by wirelessly transmitting the analog neural signals off the subject but still require a local cluster of PCs for predicting the output [1, 5]. Although this wireless approach is more inline with the ultimate BMI goal, research emphasis has been placed on shrinking the wireless acquisition hardware which still requires large immobile machines to predict trajectories [1]. Additionally, the transmission of neural waveforms has bandwidth limitations. A bandwidth bottleneck occurs as more neurons are sampled from the brain making the transmission of these signals to the digital hardware arduous and power consuming (which is contrary to the necessities of such a device) [5].

We believe that by shrinking digital hardware that computes the prediction algorithms (on the subject), the wire-

Fig. 1. Project Overview

less limitations and immobility issues are both solvable. Specifically, the 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 the fastest available wireless link. Furthermore, using on-board digital hardware to predict the trajectory removes the need for large external computers and approaches the ultimate BMI goal of patient mobility.

In this paper, we present a design for a wearable computational DSP system that is capable of processing various neural-to-motor translation algorithms. The system first acquires the neural data through a high speed data bus in order to train and evaluate our prediction models. Then via a widely used protocol, the low-bandwidth output trajectory is wirelessly transmitted to a simulated robot arm. This system has been built and successfully tested with real 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 results of the system, followed by a discussion and conclusion.

II. SYSTEM DESIGN

In order to create a successful system, it is necessary to address technical and practical aspects. To do this, we must determine how the system will function within its intended environment. Specifically for our purposes, the hardware must first receive digitized neural data during the training and evaluation of the neural-to-motor prediction models. After evaluating the output, the predicted trajectory must then be transmitted off-board through a wireless connection to a receiving computer/robot arm representing the desired control. This wireless connection must also provide the ability to remotely program and diagnose the system when

Fig. 2. Hardware Modules of WIFI DSP

being carried by a subject.

The described (WIFI-DSP) system serves as the digital portion of the overall BMI structure and is responsible for translating the neural firings into action in the external world. The first generation of this hardware was housed in a PCI slot of a personal computer and did not possess wireless capabilities [6,7]. For the second generation (discussed in this paper) we require a design that is portable, possesses wireless capabilities and is computationally fast.

Since the system needs to be portable and contain a wireless connection, we must resolve how the other system requirements are affected. First, a portable system must be light-weight and small enough for a human or primate to carry. Second, a portable system must also be self-contained and rely only on battery power. Consequently, the hardware needs to be low-power in order to extend the life of this on-board battery. The low power constraint will then influence the choice of the processor chosen since we need a low power device that can still achieve fast processing speeds. This power constraint also affects the wireless connection since it needs to be low power, yet retain enough bandwidth to transmit the output trajectory and any future data streams.

The prediction models running on the hardware platform also constrain the system design. Since most of the prediction models require the use of floating point numbers and arithmetic, we need a system that can process these floating point numbers fast enough to attain real-time model computations. Additionally, these models are sometimes large or contain multiple versions running simultaneously and therefore require large memory banks to handle the data throughput.

2.1. 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. This central component also dictates what support devices are required for the overall design. In choosing a processor for our particular system, we looked to the previous generation of the DSP board [7]. Our colleagues were able to verify that the Texas Instruments TMS320VC33 (C33) was an appropriate processor for the BMI requirements [6,7]. It was also advantageous to use this processor since there are preexisting code libraries and hardware designs.

The C33 meets our floating point and high speed 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, so that it has the ability to compute two mathematical operations in a single cycle. Even at 200MFLOPS the C33 is also well suited to our low power

Fig. 3. User Interface

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 [2].

Finally, the C33 is able to fulfill other requirements of the system. First, it is able to support a large amount of memory by providing a 24-bit address bus to read/write 16 million different memory locations. It also allows quick communication to these locations by using hardware strobe lines to directly access different memory blocks. This processor also meets the requirement of expandability since it has four hardware interrupts, two 32-bit timers, and a DMA controller that can be used for future requirements or hardware interfaces.

2.2. Wireless Communication

The second most important hardware module in our system is the wireless connection. We determined that 802.11B would be the most appropriate protocol since it is easy for our group and our collaborators to interface with. The protocol not only provides a large amount of bandwidth, it also has inspired a large code infrastructure for communication clients and servers. Additionally, by using such a widely accepted protocol, instead of developing a new one, we are able to communicate to any off-the-shelf wireless device that supports 802.11b. Essentially the system can connect to any computer or hardware device on the internet.

We designed the system to use a MA401 PCMCIA wireless 802.11B card. This PCMCIA card was the smallest and fastest card available during our design process. At the core of this card is an Intersil prism 2 chipset which is responsible for handling most of the physical layer and MAC addressing of the 802.11B protocol. The control of this chipset involves different sequences of register calls. Subsequently, these register calls help to configure, initialize, and transmit data to/from the card [10].

The PCMCIA card met the power requirements for this development stage since we have the ability to vary the power consumption from less than 100mA to 300mA depending on the bandwidth we require. It also meets our size requirements since it is slightly larger than a credit card as shown in figure 4. This device also meets the final requirement of high bandwidth since it is capable of transferring data at 10mbits/s. This high bandwidth is appropriate since it may become necessary to retrieve neural data wirelessly if a sub-dermal analog acquisition system is designed and requires data transmission through the skin.

2.3. USB

We chose the FT245BM USB FIFO device for a USB 2.0 interface. This device is small and basic to control. Additionally, it meets our low power constraint since it only uses 25mA in continuous mode and 100uA in

Fig. 4. WIFI DSP System

suspend mode. Consequently, this chip provides a range of options for data throughput requirements versus power consumption. This chip also provides an 8 mbits/s data bus for the dual purposes of data communications and system diagnosis.

2.4. Power Subsystem

There are three different power requirements in order to power the WIFI-DSP, 1.8V, 3.3V, and 5V. Texas Instruments offers a TPS70351 Dual-Output LDO Voltage Regulator that includes both 1.8V and 3.3V voltages on a single chip but only requires 5V to operate. The two output voltages are not only used by the DSP, they are also used for the other hardware modules in the system [6]. This chip also provides the power-up sequence required by the DSP once it is initialized. Additionally, having one required voltage input source is an advantage for this portable system since only one 5V battery supply is necessary.

2.5. SRAM and Expansion

The DSP has 34K by 32-bits internal high-speed SRAM. As mentioned earlier, the prediction models require more than this internal limit. Therefore, additional external 32-bit SRAM is required to connect to the C33 data bus. Unfortunately, many of the desired components and alternatives were not available during the design process. Consequently, we chose four 8-bit Cypress CY7C1049B- 15VC memory chips. These memory chips fulfill many of the requirements of this stage of development [8]. First they possess fast access time (15 ns). Second, they have low active power, (1320 mW max.) and low CMOS standby power, (2.75 mW max.) Finally, they provide easy memory expansion with their chip enable (CE) and output enable (OE) features

Having four CY7C1049B parts yields a total of 512k by 32-bits or 2MB of external memory. These four parts are incorporated using the same chip enable line connected to different bytes of the data bus, giving the appearance of a single 32-bit memory [6].

2.6. System Software

There are six layers of software in the WIFI-DSP system environment: 1) PC Software, 2) DSP Operating System (OS), 3) DSP Algorithms, 4)VHDL Code 5)802.11B Code 6)UDP protocol.

We wrote a PC console program to interface 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.

In tandem with the OS layer of the DSP, there is low-level driver code for initializing and controlling the 802.11B wireless controller. This code must interact with the DSP OS and any UDP client code or algorithms that are running simultaneously within the WIFI-DSP. Once the

Fig. 5. NLMS implementation

Fig. 6. NLMS Performance

prediction model completes an epoch or computation cycle, the program must interrupt the wireless card and transfer any required data. This process also involves creating the correct UDP packets for transmission to the appropriate UDP server (with a specific IP address).

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.

III. RESULTS

The WIFI-DSP has been fully tested via its USB interface and wireless communication in the following manner. Neural data was acquired through the USB port and used in both training and evaluation (forward) modes on the WIFI-DSP system. Specifically, the DSP was programmed to train an NLMS algorithm and upon completion, trajectory predictions were transmitted off board through the 802.11b wireless interface using a UDP client protocol. This communication occurred bi-directionally with an external laptop running as a typical UDP server.

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 average amount of time for a single prediction when computed over 90 predicted outputs is 211 usec. On a 600 Mhz PIII laptop running Matlab 6 the average prediction takes 23 msec. The factor of improvement or speed gain is around 100x for the DSP over the laptop. This result is consistent for a dedicated DSP floating point system that possesses one cycle multiplies.

The bandwidth of the wireless link (802.11b) is around 1.8 M bit/sec in continuous operation. This is comparable to what is expected on a 3Ghz Pentium laptop using the same Netgear wireless adapter with a Prism II chipset.

The current consumption is approximately 350 mA for the entire board which equates to 1750mW. Of this consumed power, over 80% or 1400mW is used by the PCMCIA wireless adapter. Overall, this is much less than the 4W previously attained by other acquisition hardware [5].

IV. DISCUSSION AND CONCLUSION

We have a working system that demonstrates LMS training on a DSP platform. Further, we were able to verify that it wirelessly transmits results to 802.11b enabled devices. However we are disappointed in the size and power consumption of the WIFI DSP board. As mentioned throughout the paper, for this development stage we relaxed

some of the constraints in order to verify the technologies and fuse them into a single system. For the next generation, we want to shrink the system in half and reduce the power consumption. Because the majority of the power is being consumed by the wireless adapter, it is necessary to find a lower power wireless link. We also want to verify the ability of the WIFI-DSP to directly communicate to analog acquisition hardware.

The WIFI-DSP system demonstrates that by shrinking digital hardware that computes the prediction algorithms (on the subject), the wireless limitations and immobility issues are both solvable. Specifically, connecting the analog and digital subsystems with a high-speed data bus is more power efficient and faster than any wireless link. Furthermore, using on-board digital hardware to predict the trajectory removes the need for large external computers and approaches the ultimate BMI goal of patient mobility.

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] Texas Instruments Incorporated, *TMS320VC33 Digital Signal Processor*, Literature Number SPRS087D, July 2002.
- [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] J. Chapin, K. Moxon, *Neural prostheses for restoration of sensory and motor function*, Boca Raton (FL): CRC Press; 2000.
- [5] 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.
- [6] S. Morrison, *A DSP-Based Computational Engine for a Brain-Machine Interface*, M.S. Thesis, University of Florida, 2003.
- [7] 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.
- [8] Cypress Semiconductor Corporation, *CY7C024/0241 and CY7C025/0251 datasheet*, San Jose (CA) USA; 1999
- [9] Texas Instruments Incorporated, *TPS70351 dual-output low-dropout voltage regulator.*, Literature number SLVS285. Dallas (TX) USA; 2000.
- [10] IEEE 802.11b Physical Layer Specifications, July 1999.