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

NeuroTalk™ An Interface for Multifunctional Neural Engineering ASICs

P. R. Troyk, D.E.A. Detlefsen, G.A.D. DeMichele, D. Kerns

Abstract—**With the availability of modern application specific integrated circuit (ASIC) design tools, simulation packages, and low-cost commercial silicon foundry processes, it is becoming increasingly easy for any laboratory, or small company, to develop a custom ASIC. For stimulation, as well as recording, chips that perform specialized functions can be designed, fabricated, and tested within a time period of 2-3 months. In many cases, the desired functionality can only be obtained by using VLSI design methods. Despite this increase in ASIC functionality, as related to neural engineering applications, there exists no common interface protocol for communicating with, and controlling, neural engineering ASICs. This would be analogous to each company that manufactures PC-based systems to have no common method of communication, e.g. USB, GBIB, RS-232, etc. While it might seem elusive, we propose the specification and development of a universal interface protocol for neural engineering ASICs. We have named this interface, NeuroTalk™.**

I. Introduction

Neural engineering is becoming increasingly dependent upon ASICs to perform a range of stimulation and recording functions. During the past decade, university and commercial laboratories have described numerous chip designs whose function and small size make them attractive to many neuroscience and neural engineering researchers. [1-12]. Unfortunately, there is no common communication and control interface shared by these and other ASIC devices, so the work of each laboratory exits in relative isolation. Within our own laboratory, we were often faced with customizing interfaces between ASICs for each new design. Therefore as common elements of these interfaces became apparent, a common interface specification and protocol emerged.

We call this interface NeuroTalk, and over the past two years have refined its function and specification so as to make it applicable to any emerging ASIC design. We now find that design of new ASICs is facilitated by having this interface in place. As a new ASIC is proposed, much of the design work needed to control or extract measured data from the chip is shortened by using modular circuits that allow the communication between the new ASIC design, existing ASICs, and PC-based controllers.

It is suggested that an evolved form of the NeuroTalk interface might become a standard that would permit ASIC designs from many laboratories to communicate with each other, and standardized software modules. While perhaps optimistic, it is beneficial to begin a dialogue in which the feasibility of a common interface protocol might be discussed within the neural engineering community. Here, we describe the current status of the NeuroTalk interface as a starting point for that dialogue.

II. The NeuroTalk Interface

A. General Description

The NeuroTalk interface is a simple, yet powerful protocol that is designed to meet the needs of communicating with ASIC logic state machines, PC-controller, microprocessors, and FPGA interface devices. NeuroTalk has two types of interface protocols, each designed to perform a specific interface need. These are named NeuroTalk–1™ (NT1) and NeuroTalk–2™ (NT2) interface and they were developed by Sigenics, Inc (Lincolnshire, IL) and the Illinois Institute of Technology (IIT, Chicago, IL) in order to facilitate communication with, and control of, ASICs designed for implantable and non-implantable neural engineering applications. NT1 is a 7-wire parallel interface that provides power (+/-), a system clock, a start of command signal (Tag), a data clock, command data, and analog I/O (NTOut) to and from the ASIC. NT2 is a 4-wire interface that provides power (+/-), encoded digital signals, and an analog I/O (NTOut), and is designed to minimize the number of wires leading to and from a remote sensing ASIC. Both NT1 and NT2 have one bi-directional analog I/O line that can be shared between multiple devices. The distinction between NT1 and NT2 lies in the manner in which digital commands are sent to the NT devices.

B. NT1 Description

The 4 lines of the NT1 digital bus are described as follows:

Tag: A rising edge of tag alerts all slaves that the start of a new instruction is beginning. A tag may interrupt a previous serial instruction that is still in progress. Any instruction interrupted by a tag is terminated. There is no hold-time requirement on Tag. After a tag pulse ends, all slaves on the bus anticipate that the next rising Clock edge will be used to latch in the first bit of a new serial instruction and that the state of Data will be stable.

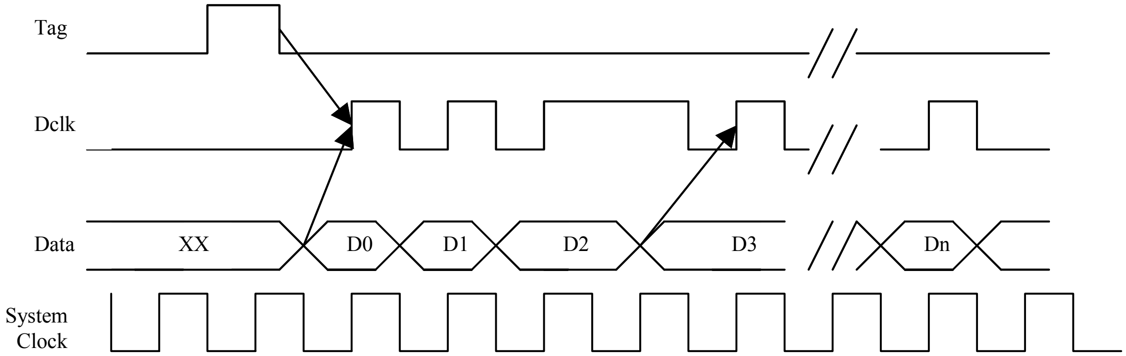
P. R. Troyk, D. Detlefsen are with Illinois Institute of Technology, Chicago, IL, USA.
G. DeMichele, D. Kerns, are with Sigenics, Inc, Lincolnshire, IL, USA.

1-4244-0033-3/06/$20.00 ©2006 IEEE.

Data: This line conveys the bit information of all serial instructions. The value of the Data line must be stable at the rising edge of a Clock pulse. The data line is disregarded while the Tag line is asserted

Data Clock: This line is used to latch in the sequential data bits provided on the Data line. It is assumed that the data line will be stable at the rising edge of Clock pulses. A single Data bit will be latched on every rising edge of Clock, so Clock and Data do not have to be periodic as long as they maintain this relationship.

System Clock: This line is used to detect transitions on the Tag, Data Clock, and Data lines. The System Clock may be a continuous duplication of the Data Clock, but does not have to be. The System Clock is understood to have a frequency at or above Data Clock.



1. The rising edge of Tag signifies a new instruction.
2. The first Data bit occurs on the rising edge of Dclk after the falling edge of Tag
3. The first Dclk edge will follow the first System clock edge after the end of a tag.
4. Data must be stable before the rising edge of Dclk
5. There is no defined time period or duty cycle for Dclk or Data
6. There is no specified length for Tag
7. Instruction lengths (in data bits) are defined by the instruction set of the slave devices

Figure 1 - NT1 interface description

Because NeuroTalk is a transmission layer protocol, Slaves are responsible for properly interpreting the serial instructions transmitted over the NeuroTalk bus. In this regard, the NeuroTalk buss is extremely non-constrained. The NeuroTalk transmission protocol does not place any constraints on the length of instructions in its scope nor does it directly provide any error checking (In wired applications, use of the NT1 protocol, over a period of 5 years, has shown it to be relatively error free, but the protocol does not inherently restrict any error correction. Error correction is used in some wireless NT designs). Instruction lengths and error checking must be embedded in the instruction itself or mutually understood by both master and slave in a NeuroTalk system. This is because no signal exists to alert the end of an instruction, only the beginning of one. Instruction interpretation (including operations, operators, instruction lengths, instruction scope, etc.) is a higher level of communication protocol that sits on top of the NeuroTalk bus and must be mutually understood by master and slave. Likewise, any data integrity schemes (Parity, CRC, Checksum, etc.) are to be implemented in the instruction definition and mutually understood by master and slave on a NeuroTalk bus. The NeuroTalk bus definition does not cover the scope of data integrity directly, but instead provides a means for flexible instruction definitions to use any scheme the master/slaves mutually understand. For example, for many designs, an address vector is transmitted as the first X number of bits in an instruction's serial stream to select which Slave should accept and execute the instruction.

In our initial development of the NT interface, we used NT1 exclusively, for all designs. Recently, we began development of a family of ASICs for use within implanted multielectrode arrays [see Kerns, et.al, and Kim, et. al, EMBS 2006]. Therefore, we were highly motivated to reduce the number of interface wires between the implanted device and a head connector or other interface module.

C. NT2 Description

NT2 is a compact form of NT1 designed to reduce the number of wired connections to the ASIC from the NT1 number of seven to the NT2 number of 4: power (+/-), digital command, analog I/O. This reduction was deemed necessary because of the extreme fabrication penalty

Data 0 = Low-to-Middle transition
Data 1 = High-to-Middle transition
Tag = Low-to-High transition
Data Clock = selective transitions,
depending upon state of Tag
Clock = Any transition

Figure 2 – NT2 digital command protocol

experienced for each additional cable wire and wire-bonded pad on the ASIC, especially when packaging within an implanted multielectrode module or array. In the NT2 interface, the Tag, System Clock, Data Clock, and Data signals are encoded onto a single wire by using level sensitive logic. The digital command signal is a three-level interface characterized by the three levels of Low, Mid, and High (~0, 1/2Vdd, Vdd; for a Vdd range of 2-5VDC). The encoding data protocol is based upon transitions between the 3 possible levels as shown in Figure 2.

D. Combined NT1/NT2 Use

Typically, a system design would utilize NT1 to form the interface between a state machine located on an ASIC, and between a microprocessor or PC-based controller. This is because the NT1 multiline interface is best suited for clocking of data from and to controllers or logic machines. The NT2 interface is best suited for single wire interface between two remote modules. Therefore, one normally produces NT1 signals from the controller, then converts them to NT2 for transmission, then converts them back to NT1 for receipt by the slave device. This conversion strategy is shown in Figure 3.

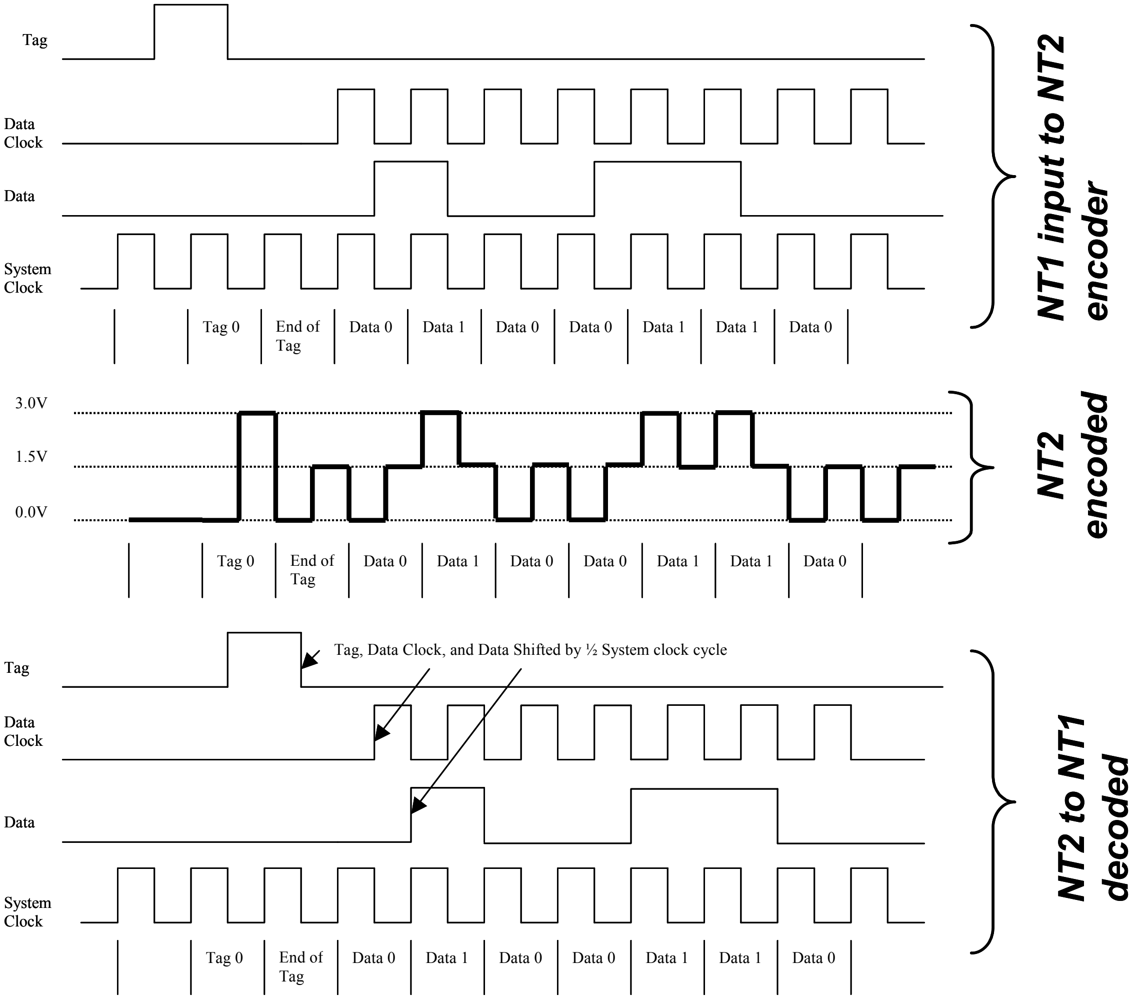


Figure 3 – NT1/ NT2 interface: typical signals

Conversion of the NT1 to NT2, and NT2 to NT1 is performed by custom hardware interface module that processes and encodes the level-sensitive logic signals. It can be easily replicated in any fabrication process, and will be the subject of an upcoming publication, although a variety of design approaches to these conversion modules are possible using the information presented in Figure 3.

III. Discussion

Although presented here in brief, the NT interface by its nature is a flexible protocol that does not place any constraints upon the content of the commands. Typically we organize device commands in the order of address, op-code, command data. For example, a stimulator chip design would have commands structured as Module Address, Channel Number for stimulation, Stimulus Amplitude, Stimulus Pulse Width [see companion paper, Troyk, et al, EMBS 2006]. For a standardized interface, only the module address would need to be preserved in the initial transmission location. By having a unique address for each NT device placed upon the buss, device conflict would be avoided. In this manner, designers can decide upon unique commands for each ASIC, while preserving the protocol for initiating and communicating the command. Several designs from multiple laboratories could share a common NT buss with the controller using the appropriate command structure for each device

REFERENCES

[1] Dagtekin M; Liu W; Bashirullah R (2001), A multi channel chopper modulated neural recording system, IEEE EMBS, Istanbul, Turkey, Oct. 23-28

[2] Diorio, C; Rajesh P.N.Rao.; (2000) Neural circuits in silicon, Nature, vol 405, June pp:891-892

[3] Fernald, K.W.; Paulos, J.J.; Stackhouse, B.A.; Heaton, R.A.;(1992) A self-tuning digital telemetry IC for use in a microprocessor-based implantable instrument Solid-State Circuits, IEEE Journal of , Volume: 27 Issue: 12 , Dec. pp: 1826 –1832

[4] Ghovanloo, M; Beach, K; Wise, K; Najafi, K; (2002) A BiCMOS wireless interface chip for micromachined stimulating microprobes, IEEE-EMBS Special topic Conference on Microtechnologies in Medicine & Biology., May 2-4, Madison WI, pp: 277 – 282.

[5] DeMichele, G.A., Troyk, P.R., (2003) Integrated Multi-Channel Wireless Biotelemetry System, Proceedings of EMBS Conference, Cancun, Mexico, September 17-21, (in press)

[6] Kim, C; Wise, K; (1997) Low-voltage electronics for the stimulation of biological neural networks using fully complementary BiCMOS circuits., IEEE trans. solid state circuits, 32, no. 10, October pp:1483 – 1490.

[7] Liu, W.; Vichienchom, K.; Clements, M.; DeMarco, S.C.; Hughes, C.; McGucken, E.; Humayun, M.S.; De Juan, E.; Weiland, J.D.; Greenberg, R.; (2000) A neuro-stimulus chip with telemetry unit for retinal prosthetic device**,** Solid-State Circuits, IEEE Journal of , Volume: 35 Issue: 10 , October pp: 1487 -1497

[8] Osborn, M.J.; Wolpert, S.; (1994) A VLSI-based biological interface for extracellular potentials Bioengineering Conference, 1993, Proceedings of the 1993 IEEE Nineteenth Annual Northeast , March 18-19 pp: 184 –186

[9] Phuoc T; Wilamowski, B; (2001) VLSI implementation of cross-coupled MOS resistor circuits IECON '01: The 27th Annual Conference of the IEEE Industrial Electronics Society, Denver, CO, Nov 29 – Dec 2, pp:1886 - 1891

[10] Qiuting Huang; Oberle, M.; (1998) A 0.5-mW passive telemetry IC for biomedical applications Solid-State Circuits, IEEE Journal of , Volume: 33 Issue: 7 , July, pp: 937 –946

[11] Troyk, P.R.; DeMicheIe, G.; Detlefsen, D.; (2002) Modular VLSI electronic design for implantable neural prostheses IEEE-EMBS, 2002. Volume: 3 , 23-26 Oct., pp: 2064 -2065

[12] Wang, G; Meijer, G.C.M; (2000) An accurate switched-capacitor amplifer with a rail-to-rail CM input range, Proc. SeSens, Dec, Veldhoven, the Netherlands, pp:703 –707.