An Advanced Demultiplexing System for Physiological Stimulation

Kelly E. Jones and Richard A. Normann,* *Member, IEEE*

***Abstract*— A CMOS very large scale integration (VLSI) chip has been designed and built to implement a scheme developed for multiplexing/demultiplexing the signals required to operate an intracortical stimulating electrode array. Because the use of radio telemetry in a proposed system utilizing this chip may impose limits upon the rate of data transmission to the chip, the scheme described herein was used to reduce the amount of digital information which must be sent to control a large quantity (up to several hundred) of stimulating electrodes. By incorporating multiple current sources on chip, many channels may be stimulated simultaneously. By incorporating on-chip timers, control over pulse timing is assigned to the chip, reducing by up to fourfold the amount of control data which must be sent. By incorporating on-chip RAM, information associated with the desired stimulus amplitude and pulse timing can be stored on chip. In this manner, it is necessary to send control information to the chip only when the information changes, rather than at the stimulus repeat rate for each channel. This further reduces the data rate by a factor of five to ten times or more. The architecture described here, implemented as an eight-channel stimulator, is scalable to a 625-channel stimulator while keeping data transmission rates under 2 Mbps.**

***Index Terms*—Electrode array, stimulation, telemetry, VLSI.**

I. Introduction

A. Existing Approaches

PREVIOUSLY, we have described the manufacture of an intracortical electrode array [1], [2]. This electrode array, dubbed the Utah Intracortical Electrode Array (UIEA), was developed to serve as an interface to the brain, for either recording electrical responses from neurons of the cortex, or for stimulation of these neurons. The UIEA consists of a 10×10 array of needle-shaped electrodes, each approximately 1400 μm in length, with a thickness at the base of approximately 80 μm, tapering to a sharp point at the tip. The individual electrodes are arranged in a square grid, with an interelectrode spacing of 400 μm. The electrodes are electrically isolated from each other at the base by a layer of glass. The tips of the electrodes are coated with platinum, and a metal pad on the backside of each electrode allows external contact to be made. Information, in the form of constant current stimulus pulses (in a stimulating application) or neural signals (in a recording application), is communicated through these pads. While we have found the geometry of this device to be ideal for cortical stimulation and recording, there are a number of other device geometries which are also being used by researchers, including planar "comb"-shaped arrays, arrays of microwires, and two-dimensional (2-D) arrays of electrodes located on a planar substrate [3]–[5].

One potential use for such a device is in a cortical neuroprosthesis such as an artificial vision system. In such a system, stimulation of an individual electrode in the visual cortex of a blind individual would produce the perception of a point of light, or phosphene [6], [7]. By properly controlling stimulation through a large array of such electrodes, a pixelized image could be produced. A video camera, mounted on an eyeglass frame worn by the user, would provide the input for such a system. Constant current pulses (typically biphasic, to balance the charge and thus help prevent damage to either the cortex or the stimulating electrode [8]), would be delivered to an individual electrode to create a phosphene. To create a maintained percept, it would be necessary to deliver not a single pulse, but rather a train of pulses. Each pulse in the train would have a total duration of 50–500 μs, and these pulses would be repeated at a rate of 20–300 Hz. The brightness of the individual phosphene would be controlled via the current amplitude, the pulse width, the pulse repeat rate, or some combination of the three [9], [10].

In the past, two methods for delivering information to and from this array have been used. The simplest consists of bonding individual wires to contact pads on the backside of the array, using either a wire bonding or soldering method [11]. The set of wires is then brought out to an external connector, where electrical connection to the group may be made. This works well for small numbers of electrodes, but becomes unwieldy when larger numbers of electrodes are used. The second technique consists of bonding a separate very large scale integration (VLSI) multiplexing/demultiplexing chip to the backside of the array [12]. A multitude of interconnects carries individual analog signals (stimulating currents or recorded neural signals) between the electrodes and the attached chip. The chip performs a time multiplex/demultiplex operation (depending upon the direction of data transfer) upon these signals, and five wires communicate this multiplexed information between the chip and the outside world.

Manuscript received November 15, 1995; revised July 22, 1997. *Asterisk indicates corresponding author.*

K. E. Jones was with the Department of Bioengineering, University of Utah, Salt Lake City, UT 84112 USA. He is now with Intel Portland Technology Development, Hillsboro, OR 97124 USA.

*R. A. Normann is with the Department of Bioengineering, University of Utah, Salt Lake City, UT 84112 USA (e-mail: normann@m.cc.utah.edu).

Publisher Item Identifier S 0018-9294(97)07596-4.

The multiplex circuitry used by us in the past was designed with simplicity in mind, and consisted of a series of CMOS passgates connecting the individual electrode lines with an external (multiplexed) signal line. An on-chip counter, using an externally supplied clock signal, scanned through the passgates in sequence, switching each one on in turn. This allows each electrode, in turn, to be read from or written to. Although this simple approach has worked well for accessing small numbers of electrodes (up to perhaps 32), it is not a workable approach for the larger arrays (625–1024 electrodes) which will be used in a functional artificial vision system [13].

A typical stimulus waveform used in such a system consists of three distinct portions: A cathodic constant current pulse, 50–500 μs in duration; a brief interphasic delay; and an anodal constant current pulse, also 50–500 μs in duration. In order to stimulate appropriately the neural elements surrounding each electrode and provide a constant percept, it is usually necessary to repeat this waveform at a rate of 30–300 Hz. A typical scheme might involve biphasic pulses of 500-μs total duration, operating at a repeat rate of 100 Hz. In this case, a total of (500 μs $\times$ 100) = 50 ms will be spent on each channel. It follows, then, that a system which relies on time-multiplexing of these signals will be able to service at most 20 channels using the given parameters. Furthermore, because the passgate system described here requires an externally supplied analog signal, it is incompatible with a strictly digital RF telemetry scheme. A more complex system of demultiplexing is necessary.

The approach used by the Center for Integrated Circuits and Sensors at the University of Michigan to demultiplex stimulating electrodes is quite simple, but effective [14]. The electrode arrays consist, in one implementation, of 16 electrodes on a chip. The associated demux circuitry is located on the same substrate. A VLSI CMOS digital-to-analog converter (DAC) current source/sink provides the stimulus current, one DAC per electrode. Each DAC (electrode) has an address, and has associated with it address decoding and a current amplitude register.

To operate this chip, the user sends a serial data stream which consists of words with the format [address:amplitude]. Circuitry on the chip decodes this serial stream, and writes the appropriate amplitude byte to the addressed DAC's amplitude register. The DAC then puts out the specified current through its electrode. All stimulus-related timing is done off-chip. To provide a typical biphasic pulse, the external controller needs to send four [address:amplitude] words: The first to turn on the cathodal current, the second to turn it off, the third to turn on the anodal current, the fourth to turn it off. There are essentially seven independent pieces of information contained in these four words: The controlled electrode address, the cathodal "on" time, the cathodal "off" time, the cathodal magnitude, the anodal "on" time, the anodal "off" time, and the anodal magnitude. (Associated dependent pieces of information are the cathodal and anodal pulse widths, and the biphasic delay.)

B. Limitations to Existing Schemes

The Michigan scheme is simple and also quite versatile, as any of the seven parameters may be varied independently, at any time. Furthermore, it is compatible with a digital RF telemetry link. However, the scheme presents some problems should one wish to use it for a 100- or 625-electrode device.

First, the area taken up by the DAC's (and their associated circuitry) is such that it may not be practical to employ one DAC per electrode on the chip. If the demux chip is to be attached via a flip-chip process to the electrode array (probably the most practical method for a large number of interconnects), then it is desirable that the demux chip be approximately the same size as the electrode array itself. Because the electrodes in our array are spaced at a fixed distance, the area available for each electrode on the chip is thus fixed. Therefore, the area which may be devoted to per-channel circuitry is also fixed. Fortunately, some thought into the patterns of stimuli to be used shows that this many DAC's would not be required. It should not, in general, be necessary to pass current through all channels simultaneously (even if stimulation at all electrode sites is desired). Rather, since the stimulus current at a given site has a short duty cycle, it is possible to time-demultiplex the pulses coming from a DAC, and use one DAC to stimulate a number of electrode sites. Making some reasonable estimates of the desired pulse widths and interpulse intervals (IPI's) one can make a conservative estimate of one DAC per four to eight electrodes.

Second, the quantity of data transferred to the chip may pose problems for a large electrode array using RF telemetered data. Performing some calculations which take into account the desired current resolution of the DAC's, required address word size, frequency of stimulation, etc., shows that driving a 625-electrode array could require a data transfer rate to the chip of over 6 Mb/s^{-1} (Mbps). For example, consider a system utilizing 625 electrodes (channels). To address a single channel would require 10 b. If 7-b D/A resolution is adequate, this results in a word length of 17 b, exclusive of any framing or error checking bits. If each of the channels is to be driven at, say, 150 Hz, the total data rate required is then four (words per biphasic pulse) * 17 (b/word) * 625 (channels) * 150 (pulses/channel/s), or 6.4 Mbps. As our stimulator is likely to require considerable power input (several mW), it would be difficult to design a telemetry system which would be able to meet the power transmission requirements while simultaneously delivering 6 Mbps of data.

A third problem with such a system is in scheduling. Suppose the off-chip controller is controlling many electrodes, each of which is being stimulated with a unique pulse width and biphasic delay. The controller has set several DAC's to the cathodal "ON" state, and has started several pulse width timers which will tell it when to turn these DAC's off. It may well happen that several timers expire simultaneously, thus presenting a scheduling conflict: Which DAC should be shut off first? Of course, it will be necessary to issue the shut down commands sequentially, which means that all electrodes shut down after the first will experience a longer than desired pulse width, the extent of which is dependent upon the data transfer rate used. Precise control of pulse width is desired, not only because it affects the perceived brightness of a phosphene, but because maintenance of charge balance between the two phases of a stimulus is required to prevent deleterious effects

upon the stimulated tissue. The Michigan group is aware of this problem, and has worked on methods of conflict resolution to be used by the off-chip controller/scheduler [15].

C. Proposed Solution

One simple way to overcome some of the problems discussed in the previous paragraphs is to utilize on-chip timers to control the timing of an individual stimulus pulse. Each DAC could have a timer associated with it. Upon chip start up, the external controller would specify a pulse width to be used for all channels. From this point on, activating an electrode would involve merely sending a single [address:magnitude] word to the chip. The timer would then take over, controlling the DAC appropriately to deliver the specified cathodal pulse width, biphasic delay, and anodal pulse width. Obviously, this would result in a fourfold reduction in data transfer rate over the scheme described originally. In addition, scheduling conflicts would be eliminated, as the on-chip timer would shut off the pulse at the appropriate time. (Scheduling conflicts which arise from the desire to turn *on* two electrodes simultaneously are relatively unimportant, since a slight delay here is tolerable. It is inaccuracy in the time between the "on" command and the "off" command which is less acceptable, due to charge balance considerations.)

Further improvements in required data transmission bandwidth are possible if one realizes that, although it may be necessary to stimulate the electrodes at rates up to 200 Hz each (for physiological reasons), a frame refresh rate for the visual information of perhaps only 30 Hz would be sufficient to create a smooth video percept. That is, with the basic scheme, it may be necessary to send information to an electrode at a rate of 200 stimuli/s, even though this information need only change at a rate of 30 times/s. Most likely, many pixels in a scene would be the same from frame to frame, and thus the actual required data update rate for an individual electrode would on the average be much less than 30 Hz.

One way to overcome this inefficiency is to implement a pixel RAM on the chip. Upon startup, the external controller would load this RAM with the current amplitude and timing information to be presented for the first pixel frame. An on-chip controller/timer would then take over, scanning through the RAM and using the information contained therein to apply stimulus pulses to the individual electrodes. From that point on, the only data which would need to be sent from the outside world is frame update information to be loaded into the RAM. If one assumes that this update is presented at, say, 20 Hz on average per electrode, as opposed to the 200 Hz required for the previous scheme, one sees that a tenfold reduction in data transfer has been achieved, in addition to the fourfold reduction achieved over the basic mode. This gets us into the realm of data transfer rates which our telemetry system is expected to be able to achieve.

In order to modulate percept intensity, amplitude modulation (AM), pulse width modulation (PWM), or frequency modulation (FM) of the stimulus waveform may be used. In our system, we have chosen to implement only the AM and PWM modalities. To some extent, these are interchangeable. However, there is usually a minimum current amplitude, below which stimulation is not possible regardless of pulse width, and a maximum current amplitude, above which stimulation may be deleterious to the tissue or the electrode. In between these limits, AM is often practiced. Once the upper current limit has been reached, however, it will be necessary to employ PWM. Because operation of any given electrode may involve switching between AM and PWM, it would be advantageous to employ a scheme which will permit this. This could be accomplished by using two separate banks of RAM, one to store amplitude information for each electrode, and one to store the respective pulse width information.

Of course, utilizing these advanced operating modes involves a tradeoff in flexibility. Whereas the basic mode can be used to send a pulse of virtually any waveform, the advanced modes will limit the user to pulses in which the cathodal and anodal phases are equal both in magnitude and pulse width. Because of this, a more flexible design would be one in which any of three modes of operation are possible: The "basic" mode, a "single pulse" mode (in which on-chip timers would control individual pulse timing), and a "continuous pulse" mode (in which on-chip RAM would contain the necessary information to send a train of pulses continuously, with only update information needed). In addition, the chip could be built to allow operation in either the standard amplitude modulated mode, or a pulse-width modulation mode in which the magnitude parameter in the [address:magnitude] word would actually refer to the pulse width, with a fixed current amplitude. The chip we have designed and fabricated to demonstrate this overall scheme meets these design criteria.

II. Chip Architecture

A. Overview

In general, the architecture designed to implement this scheme may be divided into two sections: The input section, of which there will be one per chip, and the DAC subsystems, for which there is one for each eight channels (electrodes) to be serviced. The input section consists of an input controller, which synchronizes data input with the start and stop bits, and a shift register/latch which provides a latched data and address signal from the serial input. The DAC subsystems include the DAC current source, the memory which stores waveform parameters, and the waveform timers and controllers (Figs. 1 and 2).

B. Front End (Input Section)

Serial Data Link: The chip utilizes a synchronous transmission scheme for the following reasons.

1) Asynchronous serial transmission generally requires a clock running at many multiples of the maximum bit transmission rate. A synchronous protocol does not require such a high-speed clock.
2) To achieve the desired level of pulse timing accuracy, it will be necessary to generate a clock signal off-chip and transmit this signal to the chip. Therefore, a synchronous

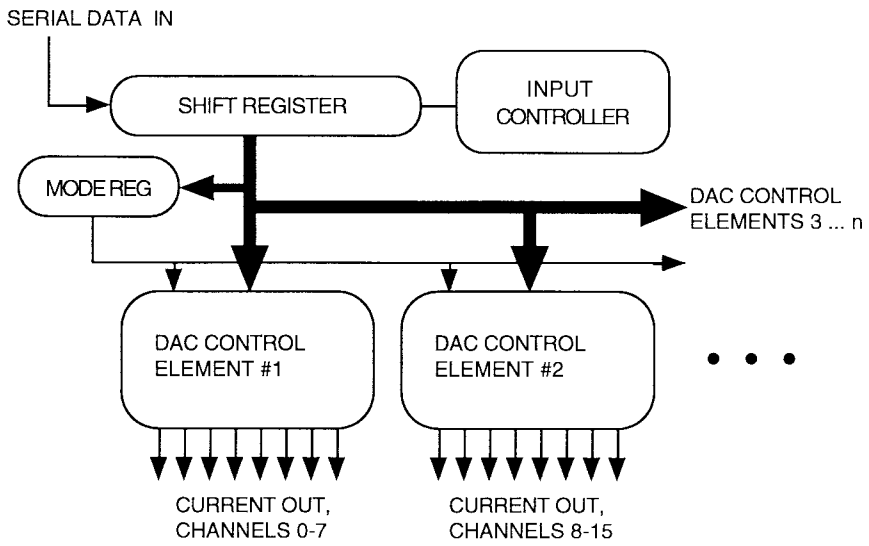

Fig. 1. Chip architecture. DAC control element architecture is diagrammed in Fig. 2.

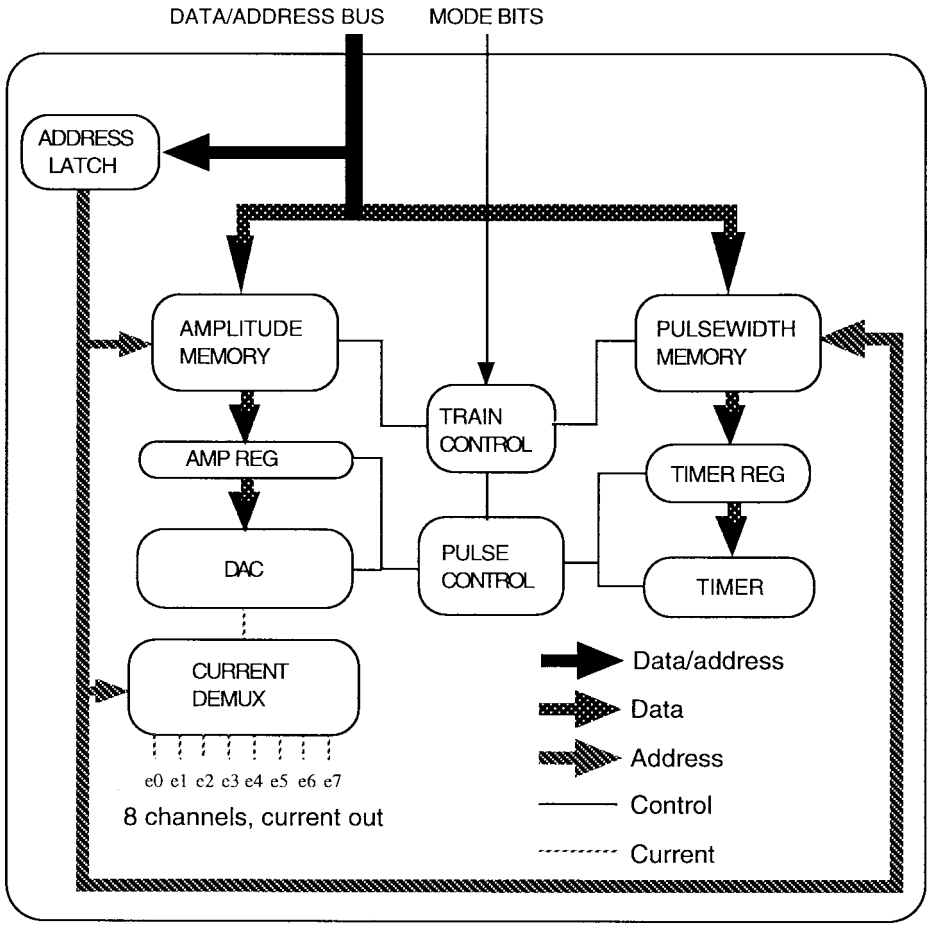

Fig. 2. DAC control element.

serial transfer scheme could easily take advantage of this shared clock signal.

Whereas, synchronous schemes generally group data in large packets utilizing error detection and correction codes, the requirements of this chip dictated that a scheme capable of sending individual data words over a (potentially) noisy link be used. Since the ultimate purpose of this chip is to create the signal necessary to generate an analog image (a visual scene), single bit errors are relatively unimportant: An occasional pixel which is incorrectly presented will not significantly change

a presented image. A more serious type of error which this system might see is a momentary "glitch" in the transmission line, in the middle of an idle (no transmission) period. It is important that a single glitch in the line will not be interpreted as the beginning of a long string of data, nor throw the chip out of synchronization with the outside world. In essence, it is necessary for the system to locate and receive a single data word which may occur at any time in a data line. Framing bits must be used which will permit accurate detection of, and resynchronization with, each transmitted word. This is accomplished as follows:

> The normal state for the transmission line is high (data 1). The beginning of a data word is indicated by a single start bit, a low (0) bit on the line. The next 14 bits sent (in the case of the test chip, which used a 7-b address and 7-b DAC resolution) are the address, least significant bit (LSB) first, followed by the magnitude, LSB first. Following this, two stop bits are sent, a zero followed by a one. By using two stop bits of opposite state, the chance that random noise on the line will generate all three start/stop bits correctly is minimal. An input controller checks for the presence of and proper location of the start/stop bits, and sets the error line (ERR) high for one cycle if not received properly.

1) Shift Register/Input Controller: Incoming data on the DATA line is clocked into a shift register, which is under the control of the input controller. The function of the controller is to detect the start bit on the data line and begin shifting in data. When the address bits have been shifted in, they are latched into the address latch. The next seven bits (the magnitude bits) are then shifted in, and the stop bits are checked for. If the appropriate stop bits are not detected at the appropriate time, the "ERR" line is set to high for one clock cycle. There is but one shift register/input controller for the entire chip, regardless of the number of channels serviced.

Mode Register: There is one mode register for the entire chip. The mode register, which is 3-b wide, can be set to one of the following modes:

001—basic mode: in this mode, every [address:magnitude] word sent to the chip causes the DAC to output a current of amplitude [magnitude], and the current is routed to the addressed channel.

010—single pulse mode: in this mode, every [address:magnitude] word sent to the chip causes a complete biphasic pulse to be sent to the addressed channel. An auxiliary address bit, denoted a_5 for the test chip, is used to determine whether the magnitude portion of the word is used to specify the current amplitude, or the pulse width. If the magnitude specifies a current amplitude, then the pulse width is taken from the pixel RAM, and vice-versa.

110—continuous pulse mode: in this mode, the train controller runs continuously, sending pulses through each of the eight channels in turn, at the specified frequency. Timing and amplitude data for these pulses are taken from the timer registers and the pixel RAM. No writes to the chip are necessary, other than an initial set of writes to initialize the RAM and registers. During this mode, any [address:magnitude] word is used to update a location in RAM, which is then used the next time the train controller accesses this memory location.

000—This is a load mode: This mode is used to initialize the RAM or registers, without causing any activation of the DAC.

It should be noted that the above implementation allows a great deal of flexibility. The chip can be operated in the basic mode, when data throughput requirements are low, but more control is desired over the specifics of the stimulation, as in testing and calibration. As the throughput needs increase, the chip can be used in either the single pulse or continuous pulse mode, depending upon the degree of control desired. Also, in any of the three modes, the stimulation can be either amplitude modulated or pulse-width modulated. In all but the continuous pulse mode, the stimulus may be frequency modulated as well.

III. Demonstration Chip

In order to demonstrate the architecture described above, a test chip was designed and fabricated (Fig. 3). In order to limit the size of the design to a TinyChip size (approximately 2 mm $\times$ 2 mm), the architecture was implemented as an 8-channel stimulator. Thus, it employs an input section and a single DAC/control subsystem. Although many of the specifics discussed here (e.g., addressing) relate strictly to the test chip, in most cases it will be readily apparent how these specifics would be modified for a chip implementing a greater number of channels.

A. Digital Components

1) Address Latch: This latch holds the address contained in the input word. The address is divided into two portions: a DAC subsystem address (a_1–a_0) which selects an individual DAC/controller system, and a channel/special address (a_6–a_2) which selects a channel or register within that system. The channel/special address is latched into the subsystem's address register whenever the subsystem is specifically addressed by a_1–a_0.

2) Amplitude Register: There is one amplitude register associated with each DAC system. This register holds the value which specifies the current to be output through the DAC. The register is 7-b wide, the highest bit of which specifies the sign of the current. A signal from the pulse controller will invert this bit in order to achieve the second (opposite sign) phase of a biphasic pulse.

3) Timer Registers T_1, T_2, T_3: Three 7-b timer registers hold the values for the cathodal/anodal pulse width, the biphasic delay, and the interpulse interval, respectively. Registers T_1 and T_2 may be written either from pixel RAM or from the data bus, whereas T_3 is written only from the data bus.

4) Timer: This is a loadable, seven bit countdown timer which is loaded from one of the three timer registers. This timer was designed to operate at dual frequencies of 250 and 1.953 kHz (250 kHz divided by 128). The 250-kHz clock is used to time the pulse width and biphasic delay, allowing resolution of 4 μs. The slower clock speed is used to time the

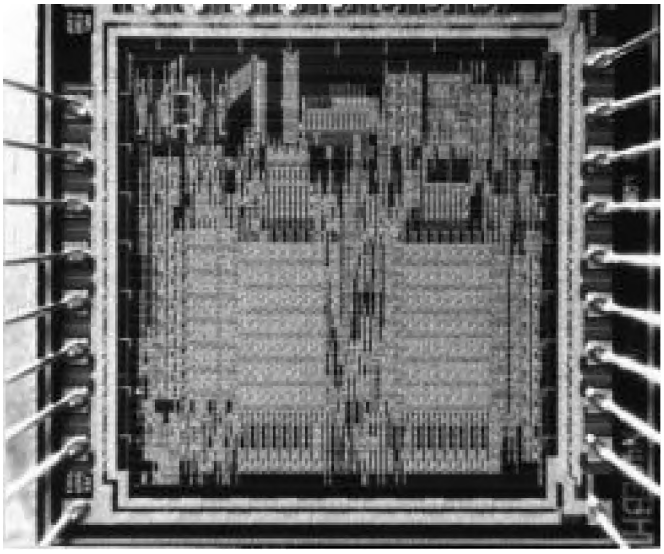

Fig. 3. Photo of demonstration chip.

interpulse interval. In the general architecture, the 250-kHz clock would be derived from the chip master clock (which in turn would be derived from the RF carrier frequency in our proposed telemetry system) by a divider. The slower clock (1.953 kHz) would be derived from the 250-kHz clock by a 7-b divider. On the demonstration chip, the chip master clock was chosen to be 250 kHz (because of the low serial transmission rate required for an 8-channel stimulator), thus, this clock served directly as the timer clock. Due to space limitations, the slow clock was generated externally, rather than providing a 7-b divider on the chip.

5) Pulse Controller: The pulse controller controls the output of a complete biphasic pulse through the DAC. It is initiated either by a signal from the train controller, or by a data write while in the single pulse mode. It loads the timer with the pulse width (register T_1), turns on the DAC, and awaits a time-out signal from the timer. Upon time-out, it turns the DAC off, loads and restarts the timer with the biphasic delay (register T_2), and again awaits time-out. Upon time-out, it changes the sign of the value in the magnitude register by inverting the highest (sign) bit, then loads the timer once again with T_1, restarts the timer, and awaits time-out. Upon time-out, the controller switches the DAC off, and awaits reinitiation.

6) Pixel Ram: There are two 8-byte $\times$ 7-b arrays of RAM which store pixel information. The first of these stores an amplitude for the DAC, and the second holds pulse width values for the timer. These are loaded from the data bus, and send their output to the magnitude register or the pulse width Timer register T_1, respectively.

7) Train Controller: The train controller controls the sending of a continuous train of pulses to all eight channels serviced by a single DAC. The controller is running whenever the chip is in the continuous pulse mode, mode 110. The controller functions by setting a channel counter to zero, loading a magnitude and a pulse width from the RAM into the magnitude and T_1 registers, respectively, and then starting the pulse controller. When the pulse controller signals the train controller that a complete biphasic pulse has been sent to the selected channel, the train controller increments the counter, reloads the registers from RAM, and again starts the pulse controller. This continues until all eight channels have been serviced. At this time, the train controller loads the timer with the interpulse interval value stored in T_3, and waits for time-out. During this time interval, the timer is instructed to count at 1/128th of the normal speed, by using the "SLOW" clock. Upon time-out, the controller begins this entire cycle again.

8) Addressing: A 7-b address is used to access all channels and registers on the chip. The seven bits are used as follows.

a_1a_0 These bits select an individual DAC on the chip. Of course, this chip only contains one DAC, but two bits were provided to simulate a more general architecture. The special address $a_1a_0 = 11$ is used to access all DAC's in the architecture. This allows, for example, every register on the chip to be simultaneously written with some default value for initialization, if so desired. The single DAC implemented on this chip is addressed by $a_1a_0 = 00$.

$a_4a_3a_2$ These three bits are normally used to address one of eight individual channels (electrodes) within a DAC subsystem. This "normal" addressing occurs when $a_6 = 0$. If $a_6 = 1$ ("special" addressing), bits a_5–a_2 are used to specify other options (discussed below).

a_5 This determines whether the [magnitude] portion of the data word refers to a current amplitude or a pulse width, in an advanced mode. (In the basic mode, the magnitude always refers to the current amplitude, as pulse-width is not controlled by the chip.) If a_5 is zero, the magnitude is construed to be a current amplitude, and is written to both the amplitude register and the amplitude portion of RAM. If $a_5 = 1$, the magnitude is construed to be a pulse width, and is written to both the timer register T_1 and to the pulse width portion of RAM.

a_6 This bit determines whether the address bits a_5–a_0 are to be construed as normal or special. If a_6 is 1, the address is special, and the following special addresses are decoded: (Note that DD refers to the address of the specific DAC subsystem referenced, X = do not care.)

- 1T10XDD Used to write to the timer registers. If $T = 0$, T_2 is written; if $T = 1$, T_3 is written.
- 1X110XX Used to write data to the mode register. Since the mode register is only 3-b wide, only the three lowest order data bits are written to the register.
- 1X111XX Any write to this address resets the entire chip.
- 1M00XDD This address is used to write (initialize) an entire bank of RAM with a single value. If $M = 0$, the amplitude RAM is written, and if $M = 1$, the pulse width RAM is written to.

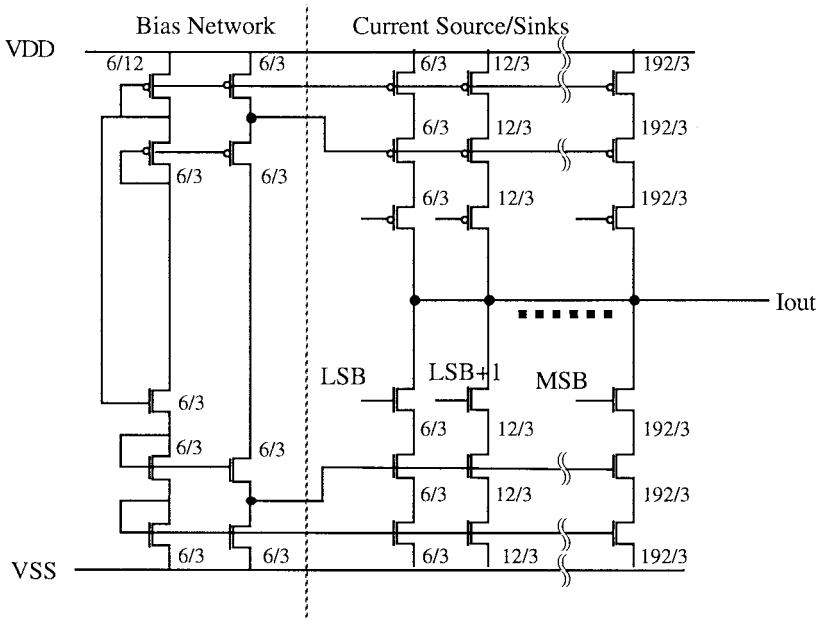

Fig. 4. Schematic diagram of the constant current DAC circuit. Numbers indicate the W/L ratio for each of the pictured transistors.

B. Subsystem: Analog Elements

1) DAC: The DAC is a simple current mirror device based upon a scheme outlined in Allen and Holberg [16]. This DAC is schematically diagrammed in Fig. 4. It can be divided into two parts, a bias section and a current source/sink section. The bias section provides four bias voltages: two for the NMOS current sinks, and two for the PMOS current sources. A design was chosen which would permit current sourcing/sinking as close as possible to the power rails. The circuitry and transistors are sized such that the bias voltages provided are fairly close to the power rails, ensuring that the source/sink transistors will remain in saturation until the output voltage gets quite close to the rails. This allows for relatively low voltage supplies, (always desirable in an implanted device), and also creates a current source with a higher compliance for a given supply voltage.

The current source/sink section consists of sets of transistors with currents mirrored from the bias section. The transistor widths are scaled such that the transistor gated by (LSB + 1) supplies twice the current as that gated by LSB, and so on. In order to prevent scaling errors, the larger transistors are implemented as multiple instances of unit sized transistors, rather than a single large transistor. These multiple instances are arranged in a common centroid formation in order to reduce errors associated with variation of parameters across the chip. In series with each set of source/sink transistors is a single transistor switch, gated from the digital portion of the chip, which enables or disables that particular source/sink.

2) DAC Demultiplexer/Passgates: This is a set of eight CMOS passgates, selected by a three-to-eight decoder which decodes the three-channel address lines. The passgates determine which of the eight channels is active, and also connect all of the unused channels to the exhaust line.

3) Required Signals: The architecture, thus, described (as implemented on the test chip) requires eight external connections, exclusive of the connections to the individual electrodes.

V_{DD}	(nominally +3 V).
V_{SS}	(nominally −3 V).
CLOCK	The timers on this chip were set up to utilize a 250 kHz clock.
DATA	This is the serial data line.
ERR	This line is an output, it is set to HIGH by the input controller to indicate that an error has been detected in the input serial data stream. In future versions of the chip, this line could also be used to indicate other types of fault conditions on the chip, such as over-voltage conditions on electrodes, or information from a chip "watchdog."
EXHAUST	This line is used to exhaust built-up charge from unused electrodes. This should be connected to a ground (0 V) node. In future versions of the chip, this could be replaced by an on chip current source/sink operating at 0 V ($[V_{DD} + V_{SS}]/2$).
RESET	This pin resets the three internal controllers (input controller, pulse controller, train controller). In future versions of the chip, a power-on-reset (POR) could be implemented.
SLOW	Clock-timing of the IPI requires a division of the primary timer clock by a factor of 128. Although this is currently done by an off-chip divider, future versions of the chip could incorporate the divider on-chip.

It can be seen that the first five of these signals are mandatory for any implementation of this general architecture,

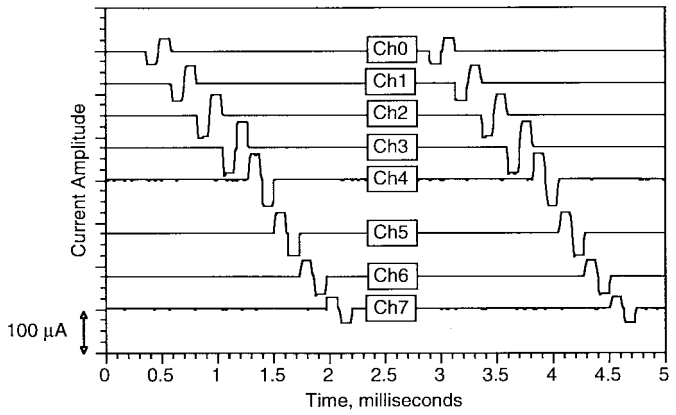

Fig. 5. Eight channels of current output in continuous pulse mode. The eight channels were each instructed to output a biphasic pulse, with each phase having a pulse width of 100 μs, and a brief (10 μs) biphasic delay. Channels 0–3 were instructed to output cathodic-first pulses of 60, 80, 100, and 120 μA, and channels 4–7 were instructed to deliver anodal-first pulses of 120, 100, 80, and 60 μA, respectively. Resolution is limited by the resolution of the digital oscilloscope used. Traces are displaced vertically by an arbitrary amount for the purpose of clarity.

whereas, the last three could easily be eliminated on future versions of the chip.

IV. Chip Testing

The chip was fabricated using the Orbit 2.0 μm n-well analog CMOS process, brokered by the MOSIS service. The operation of the chip in the digital realm was verified in benchtop testing, and all functions operated as designed. The chip functioned properly at external clock speeds up to 5 MHz, above which errors (which were manifest as timing errors) were observed. A typical output plot of the eight output channels is shown in Fig. 5. Each of the eight channels was connected through a 10-kΩ resistor to ground, while the chip was powered at $\pm$3 V. Although the measured parameter was voltage, these voltages have been converted to the corresponding currents for the ordinate axis of Fig. 5. The chip was operated in the continuous pulse mode, with the pulse width for all channels set at 100 μs, and channels 0–3 were instructed to deliver cathodal-first pulses of 60, 80, 100, and 120 μA, whereas channels 4–7 were instructed to deliver anodal-first pulses of 120, 100, 80, and 60 μA, respectively.

V. DAC Characterization

In order to evaluate the suitability of the DAC for use as a current source for physiological stimulation, four characteristics were examined.

1) DAC Absolute Output as a Function of V_{DD}: The current put out by this chip is, in essence, a mirrored current of some reference current. Generally speaking, it is difficult to build an absolute current reference in CMOS VLSI. Although it is possible to build a voltage reference which is more or less constant with respect to temperature, V_{DD}, etc., creating a current reference from this would require a precision resistor. It is also difficult to build precision resistors in standard VLSI processes. This leaves us with few choices for generating the current reference necessary. One option might be to generate the reference current off-chip. This, of course, would require an extra lead, in an application where the goal is a minimal external lead count, as well as being incompatible with a telemetry system. Another approach would be to make the on-chip reference a strong function of V_{DD}. In this way, the absolute current output could be calibrated by using slight adjustments to the chip V_{DD} level. This was the approach chosen for this chip.

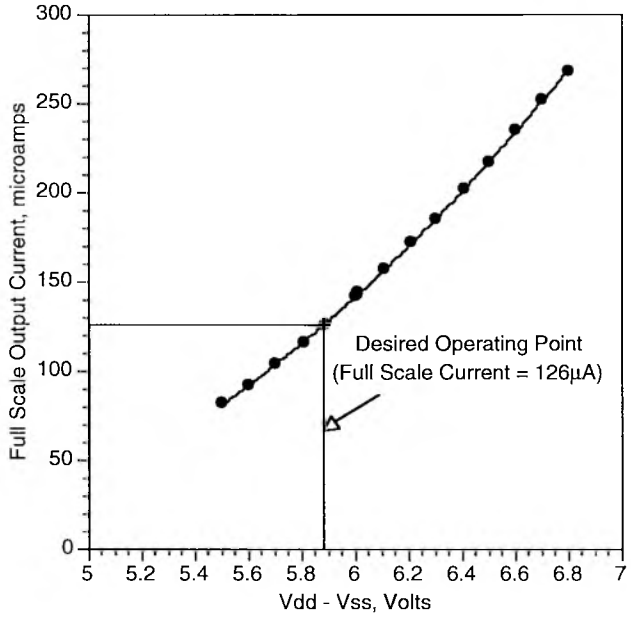

Fig. 6. Full-scale output current of DAC as a function of operating voltage.

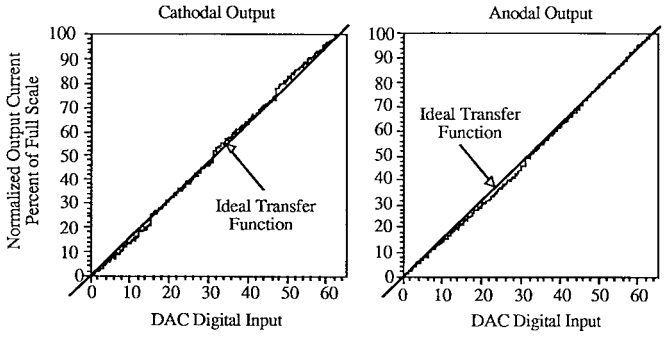

Fig. 7. Linearity of DAC transfer functions.

HSPICE modeling, using an estimate of the process parameters, was used to size the transistors used in the on-chip current mirror. In order to verify the actual relationship between V_{DD} and the output current, the full scale output of the DAC was measured while the input voltage to the chip V_{DD} was varied. The results as measured on the chip are shown in Fig. 6.

2) The Performance of the DAC in Terms of Accuracy, Differential Accuracy, and Monotonicity: The chip, operating in the basic mode, was instructed to output a current which increased in amplitude in 1-b steps. The measured results for both anodal and cathodal sweeps are indicated in Fig. 7. Although it can be seen that both the absolute and differential accuracy of the DAC is off by as much as two LSB's, the response is monotonic throughout its range.

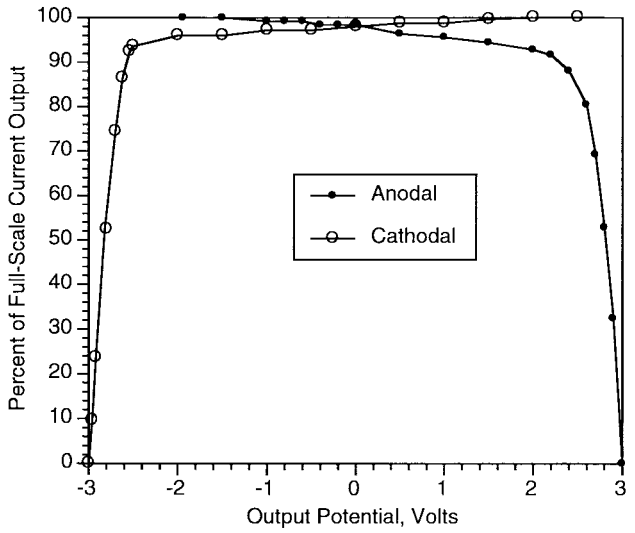

Fig. 8. DAC output compliance. DAC current output as a function of the output voltage.

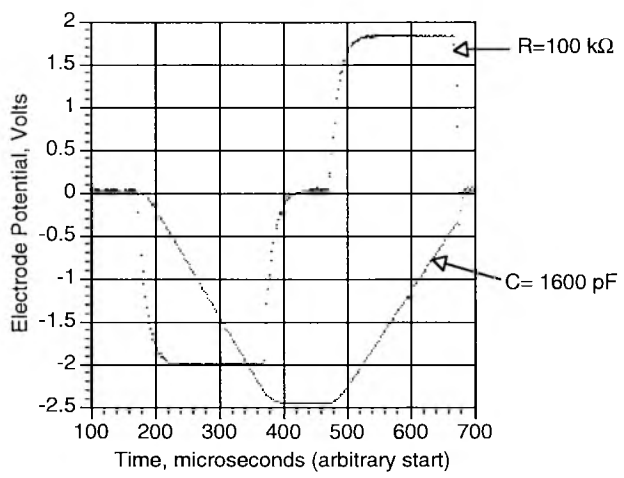

Fig. 9. Demonstration of charge balance and charge exhausting performance. Voltage drop across a resistive and a capacitive load for a constant current, biphasic pulse.

3) DAC Compliance: The compliance of the current sources/sinks were evaluated by varying the voltage present at the output node, and measuring the resulting current. Results are indicated in Fig. 8. In normal operation, the stimulating electrode will have a resting potential near the midpoint of the power rails (±3 V as implemented here), thus, the flatness of the source or sink transfer function should be evaluated between 0 V and the appropriate rail. It can be seen that the cathodal sink varied by less than 2% over this range, to within approximately 0.5 V of the negative rail. The anodal source performed less well, varying by about 6% within 1 V of the rail. This performance will be improved in future versions by a resizing of the DAC transistors.

4) DAC Charge Balance/Exhaustion: The balance between anodal and cathodal currents, and the efficacy of the exhausting scheme are demonstrated in Fig. 9. A constant current, biphasic pulse, of amplitude 20 μA, pulse width 200 μs, and biphasic delay of 10 μs was delivered by the DAC through a load. In one case, the load used was a 100-kΩ resistor. As can be seen, the anodal phase is lower in amplitude than the cathodic phase, due to the differing compliance as discussed in the previous paragraph. In the other case, the load used was a 1600-pF capacitor, which has an equivalent impedance of 100 kΩ at $f = 1000$ Hz. The imbalance of the two phases is evidenced here by a failure of the load voltage to return exactly to zero at the end of the anodal pulse. However, immediately after the pulse ends, the electrode is "exhausted" by the internal circuitry, and the excess charge on the electrode is very quickly reduced to zero.

VI. Discussion

A. Tradeoffs Between Operating Modes

Although the more advanced operating modes of this chip do indeed meet the design specifications while still retaining reasonable data throughput requirements, there are tradeoffs involved that might motivate the selection of one mode of operation over another, for a given situation. For instance, when operating in the "single pulse" mode, the shape of the pulse is constrained to a biphasic square wave pulse, with equal cathodal and anodal pulse widths, and equal but opposite current amplitudes for both phases. These restrictions are not present in the basic mode, which could used to produce stepped-current pulses, for example, or biphasic pulses with differing pulse widths/magnitudes between the phases. When operated in the continuous pulse mode, even more limitations are present. For example, each channel is stimulated using the same IPI, which eliminates the possibility of stimulating different channels at different frequencies.

B. Scalability to a 625-Channel Device

Previous psychophysical experiments [13] suggest that a useful visual neuroprosthesis might require on the order of 625 pixels, or channels. As such, it is important that the schemes described herein be scalable to this degree, and an example of the required scaling follows.

To handle 625 channels, while keeping a "packing ratio" of one DAC per eight channels, would require 79 DAC's. The addressing for these DAC's would thus require seven bits to select a DAC, plus three bits to select a channel within the DAC, plus one bit to specify whether the [magnitude] is a pulse width or a current amplitude, for a total of 11 address bits. ("Special" addresses, as discussed in this paper, could be handled by using some of the unused 7-b DAC addresses, as only 79 of the 128 possible 7-b DAC addresses will be used.) Adding seven magnitude bits and three start/stop bits thus, results in a total word length of 21 bits.

In order to stimulate each of the 625 channels at a rate of 150 Hz, while operating in the "basic" mode, would thus require an input data rate of 7.9 Mbps. To operate in the single pulse mode would require a data rate of just one fourth this, or just under 2 Mbps. Further speed reductions, perhaps to 500 kbps, are possible if the "continuous pulse" operation mode is used. Perhaps the most reasonable compromise would be to run the input section at a clock speed of 2 MHz, which would

TABLE I
ARCHITECTURE SCALABILITY

Component	Size on 8 channel test chip, mm^2	Size per channel, 2.0 μm process, mm^2	Size per channel, 0.8 μm process, mm^2
Front end (Input controller, etc.)	0.46	(note 1)	(note 1)
Test pads and bus, etc. (31 pads total)	1.90	0.061	0.061 (note 2)
Analog components (DAC and passgates)	0.14	0.018	0.018 (note 3)
Digital Components	2.30	0.288	0.058 (note 4)
Total Area	4.8	0.367	0.137

(1) The input section has not been considered here, as only 1 input section is required for the entire chip, regardless of number of channels. Although the input section occupies a substantial proportion of the 8 channel chip, its relative size on a 625 channel chip will be negligible.
(2) The pads area must remain large enough to facilitate bonding between the chip and electrode array, thus the pads are not scaled.
(3) Because the crucial characteristics of the analog portions of the chip are dependent upon their size, these portions will not be scaled.
(4) An 80% reduction in area has been assumed for scaling this circuitry from 2.0 μm to 0.8 μm.

allow sufficient data transfer rates for running in the single pulse mode. A simple divide-by-8 circuit could provide the 250-kHz clock required for timing of the stimuli pulse widths.

To scale the circuitry to a 625-channel device would require a process scaling as well, as the 2.0-μm process does not produce sufficient circuit density to fit the digital circuitry into the allotted space. The goal is to fit the per-channel circuitry into a per-channel space allotment of 400 μm $\times$ 400 μm, or 0.160 mm^2. Table I shows the sizes of the various elements, and how they might be scaled to achieve the desired density. As can be seen, the total area required for the circuitry (0.137 mm^2) easily falls within the per-channel allotment of 0.160 mm^2.

It is also possible to make a rough estimate of the power requirements for a scaled up version of the demonstration chip. As most of the power consumed by a CMOS chip is used to drive gate and stray capacitances [17], these capacitances can be used to estimate power usage from the formula

$$P = nCV^2fx$$

where

- P power;
- n number of transistors on chip;
- C average stray and gate capacitance per transistor;
- V voltage (V_{DD});
- f frequency at which chip is driven;
- x fraction of transistors (on average) which transition each cycle.

The demonstration chip, which contained roughly 5000 transistors, had a typical nodal capacitance of 23 fF. For V_{DD} = 6 V and f = 250 kHz, and assuming a value of 0.35 for x, this equation yields a value of P = 360 μW. This agrees reasonably well with the measured value of 450 μW. If the chip were fabricated from 0.8-μm technology, a reasonable estimate of the average nodal capacitance would be 10 fF. Although the input section of the chip would need to be operated at a higher speed in order to accommodate data transfer, it is likely that the majority of the functional blocks on the chip would run at 250 kHz, the clock used by the timers. Finally, the number of transistors can be approximated by taking a ratio of the desired number of output channels, thus $n_{(625/8)} \times 5000 = 390\,625$. Plugging these numbers in yields a value of 11 mW to power the chip. It should be noted, however, that the digital circuitry has not been optimized for minimal power consumption. Doing so could result in significant power savings. Further reductions in power usage could be accomplished by using separate supplies for the DAC and the digital section. Changing the voltage used for the digital section from 6 to 3.3 V would reduce power roughly by a factor of three.

In addition to the power used to run the chip, the stimulus current used in stimulation would require approximately 7 μW/channel, or an additional 4.5 mW for a 625-channel device. This could result in a total power dissipation of up to 16 mW (assuming no power optimization). This power is well within the capability of the telemetry circuitry now under development [18]. Whether this amount of heat can be dissipated safely within the brain is a topic for further study.

VII. CONCLUSIONS

An architecture has been designed that will permit generation and control of biphasic constant current stimulating pulses destined for delivery to an array of intracortical electrodes. The architecture is particularly suited to use with an RF telemetry system. Three modes of operation are provided, each mode offering a different tradeoff between required data transfer rate, and degree of control over stimulus parameters. In all modes, the stimulus may be pulse-width or amplitude modulated. Furthermore, the architecture is scalable, in terms of data throughput, required size, and power, to an implementation

requiring up to 625 stimulating channels. The architecture has been optimized to reduce the total external lead count necessary to control the device, and to minimize the data input necessary to control the device. An eight-channel version of the architecture has been fabricated in 2.0-μm CMOS, and was found to perform satisfactorily.

REFERENCES

[1] K. E. Jones, P. K. Campbell, and R. A. Normann, "A silicon/glass composite intracortical electrode array," *Ann. Biomed. Eng.,* vol. 20, pp. 423–437, 1992.

[2] P. K. Campbell, K. E. Jones, R. J. Huber, K. W. Horch, and R. A. Normann, "A silicon based three dimensional neural interface: Manufacturing processes for an intracortical electrode array," *IEEE Trans. Biomed. Eng.,* vol. 38, no. 8, pp. 758–768, 1991.

[3] N. A. Blum, B. G. Carkhuff, H. K. J. Charles, R. L. Edwards, and R. A. Meyer, "Multisite microprobes for neural recordings," *IEEE Trans. Biomed. Eng.,* vol. 38, no. 1, pp. 68–74, 1991.

[4] D. Jaeger, S. Gilman, and J. W. Aldridge, "A multiwire microelectrode for single unit recording in deep brain structures," *J. Neurosci. Methods,* vol. 32, pp. 143–148, 1990.

[5] J. Kruger and M. Bach, "Simultaneous recording with 30 microelectrodes in monkey visual cortex," *Exp. Brain Res.,* vol. 41, pp. 191–194, 1981.

[6] M. J. Bak, J. P. Girvin, F. T. Hambrecht, C. V. Kufta, G. E. Loeb, and E. M. Schmidt, "Visual sensations produced by intracortical microstimulation of the human occipital cortex," *Med. Biol. Eng., Comput.,* vol. 28, pp. 257–259, 1990.

[7] E. M. Schmidt, M. J. Bak, F. T. Hambrecht, C. V. Kufta, D. K. O'Rourke, and P. Vallabhanath, "Feasibility of a visual prosthesis for the blind based on intracortical microstimulation of the visual cortex," *Brain,* vol. 119, pt. 2, pp. 507–522, 1996.

[8] J. C. Lilly, J. R. Hughes, E. C. Alvord, and T. W. Garkin, "Brief noninjurious electric waveforms for stimulation of the brain," *Sci.,* vol. 121, pp. 468–469, 1955.

[9] W. H. Dobelle and M. G. Mladejovsky, "Phosphenes produced by electrical stimulation of human occipital cortex, and their application to the development of a prosthesis for the blind," *J. Physiol.,* vol. 243, pp. 553– 576, 1974.

[10] G. S. Brindley and W. S. Lewin, "The sensations produced by electrical stimulation of the visual cortex," *J. Physiol.,* vol. 196, pp. 479–493, 1968.

[11] C. T. Nordhausen, P. J. Rousche, and R. A. Normann, "Optimizing recording capabilities of the Utah intracortical electrode array," *Brain Res.,* vol. 637, pp. 27–34, 1994.

[12] K. E. Jones and R. A. Normann, "Demultiplexing of an intracortical electrode array: Circuitry and interconnect techniques," unpublished.

[13] K. Cha, K. Horch, and R. A. Normann, "Simulation of a phosphene based visual field: Visual acuity in a pixelized vision system," *Ann. Biomed. Eng.,* vol. 20, pp. 439–449, 1992.

[14] S. J. Tanghe and K. D. Wise, "A 16-channel CMOS neural stimulating array," *IEEE J. Solid-State Circuits,* vol. 27, no. 12, pp. 1819–1825, 1992.

[15] "Stimulating electrodes based on thin-film technology," Solid State Electronics Laboratory, Bioelectric Sciences Laboratory, Dept. Electr. Eng. and Comput. Sci., Univ. Michigan, Quarterly Rep. #8, Oct.–Dec. 1988.

[16] P. E. Allen and D. R. Holberg, *CMOS Analog Circuit Design.* Fort Worth, TX: Holt, Rinehart, and Winston, 1987, p. 701.

[17] R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI Design Techniques for Analog and Digital Circuits.* New York: McGraw-Hill, 1990, pp. 597–598.

[18] M. R. Shah, R. P. Phillips, and R. A. Normann, "A transcutaneous power and data link for neuroprosthetic applications," in *Proc. 15th Annu. Int. Conf. IEEE Engineering in Medicine and Biology Society,* 1993, pp. 1357–1358.

Kelly E. Jones received the B.S. degree in chemical engineering, in 1985, from the Ohio State University, Columbus, and the Ph.D. degree in bioengineering, in 1995, from the University of Utah, Salt Lake City. His dissertation concerned the development of electrode arrays and VLSI circuitry as part of a system for forming an electrical interface to the cerebral cortex.

From 1985 to 1988, he was a Product and Process Development Engineer for Corning, Inc., Corning, NY, developing applications for glass and ceramic materials. He is currently a Senior Design Engineer for Intel, Hillsboro, OR, designing high-speed microprocessors.

Richard A. Normann (M'88) received the BS, MS, and Ph.D. degrees in electrical engineering from the the University of California, Berkeley.

He joined the staff of the National Institutes of Health, Bethesda, MD, in 1974 and remained there until 1979, when he moved to the University of Utah, Department of Bioengineering, Salt Lake City, where he presently serves as Chairman. He also holds Research and Adjunct Professorships in the Departments of Ophthalmology and Physiology in the School of Medicine at the University of Utah. His research interests are information processing in the vertebrate visual system and neuroprosthetics (creating interfaces to the nervous system).