33rd Annual International Conference of the IEEE EMBS
Boston, Massachusetts USA, August 30 - September 3, 2011

Analysis and Design of Data Transmission Protocol for 1024-channel Retinal Prosthesis

Kuanfu Chen, Ming-Hsien Tsai, Fu-Lung Hsueh, and Wentai Liu

***Abstract*— Epiretinal prostheses deliver electrical pulses through a great number of stimulation electrodes to generate visual perception. Since a large amount of stimulation parameters have to be sent into the eye wirelessly, the transmission efficiency and data loss are critical design factors. While the error rate of wireless transmission might not be perfect due to the limited power dissipation allowed in the eyeball, data loss can be reduced by altering the size of data packets. In this paper, the correlation between packet length and data loss are analyzed, and we find that more than 10 times of reduction in data loss can be achieved by shortening the data packets. The costs and benefits of using error-correcting codes (ECC) are also evaluated. A data packet protocol for 1024-channel retinal prosthesis as well as the corresponding receiver circuitry are then designed based on the results of the analyses. The circuit occupies 330μm $\times$ 262μm using 0.18μm CMOS technology. With the flexibility provided by the protocol and receiver circuitry, data packet configuration (size and ECC on/off) can be adaptively adjusted to optimize for real-time wireless channel error conditions.**

I. Introduction

Functional electrical stimulation (FES) has been utilized in several commercially available medical products such as cochlear implants, artificial cardiac pacemakers, and deep-brain stimulators. The retinal prosthesis is a new application in this field. It applies electrical pulses on the retina through an array of electrodes to induce artificial visual perception. Patients who lose their vision due to degeneration of the light sensing photoreceptor cells may benefit from the prostheses by its ability to restore partial eyesight. Research on retinal prostheses has made great progress in the past few years. The retinal prosthesis device built by Second Sight Medical Products, Inc. (Sylmar, California, USA; www.2-sight.com) has got CE Mark, and may be commercially available in the European Union later this year.

An epiretinal prosthesis is usually composed of external and internal units, as illustrated in Fig. 1. Images captured by the eyeglass-mounted video camera are translated to stimulation parameters and encoded using a data packet protocol. These data packets are then transmitted to the internal unit through wireless data telemetry. The system-on-a-chip (SoC) implanted in the eyeball decodes the received data packets and converts the stimulation parameters into electrical current stimuli to the electrode array sitting on the retina.

This work was partially supported by the National Science Foundation through BMES-ERC and by the Department of Energy through the Artificial Retina Project.

K. Chen and W. Liu are with the University of California at Santa Cruz, Santa Cruz, CA 95064 USA e-mail: kuanfu@soe.ucsc.edu

M. Tsai and F. Hsueh are with the Taiwan Semiconductor Manufacturing Company, Ltd., Hsinchu 300, Taiwan

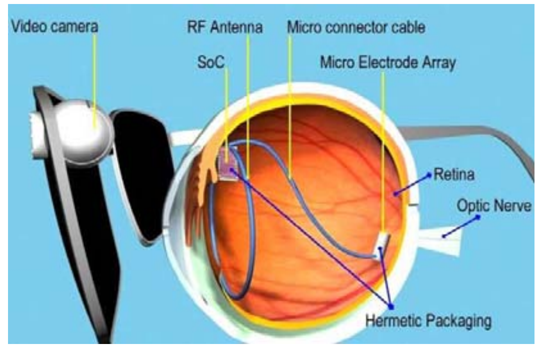


Fig. 1. A system overview of the epiretinal prosthesis. The data transmission protocol is used in the wireless telemetry from the video camera to the SoC inside the eyeball. (Image courtesy of NSF-ERC-BMES)

Contrary to other implantable FES devices, retinal prostheses usually have a larger number of electrodes. The number of electrodes is one of the major specifications in determining the quality of the artificially generated visual perception. There are 60 electrodes in the epiretinal prosthesis currently used in clinical trials, and more electrodes will be expected in the next generation of retinal prostheses. Research results reported in [1] showed that at least 1000 pixels are required to restore important visual functions, such as face recognition and reading. With more and more electrodes in the internal unit, the amount of data required to transmit through wireless telemetry rises significantly. Therefore, data transmission efficiency becomes a critical factor in designing retinal prostheses.

In the previous papers [2], [3], we reported the design consideration of the digital controller in the SoC for retinal prostheses. Features of the controller, such as the on-chip address generation and configuration mode, reduce the amount of data required for each stimulation channel. In this work, we analyze the data loss due to wireless transmission error in various conditions. The simulation results also show that the rate of data loss can be significantly reduced by changing the data packet length and adding error-correcting code (ECC). Based on the analysis, the data transmission protocol is designed, and the corresponding decoding circuit at the receiver SoC is built.

This paper is organized as follows: transmission efficiency is analyzed in Section II; data packet protocol is presented in Section III; followed by the receiver circuit implementation in Section IV. In Section V we present our conclusions.

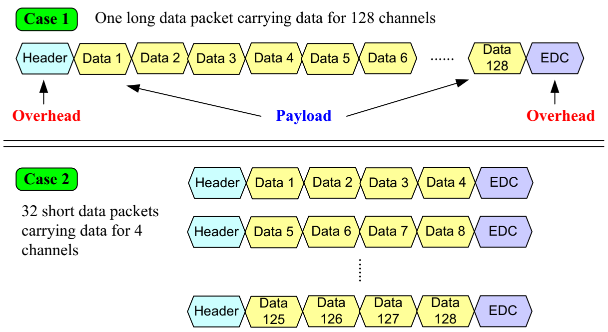


Fig. 2. Data packet for wireless transmission composed of three parts: header, data, and error detection code. Data can be packed into one long packet or several short packets.

II. Transmission Efficiency Analysis

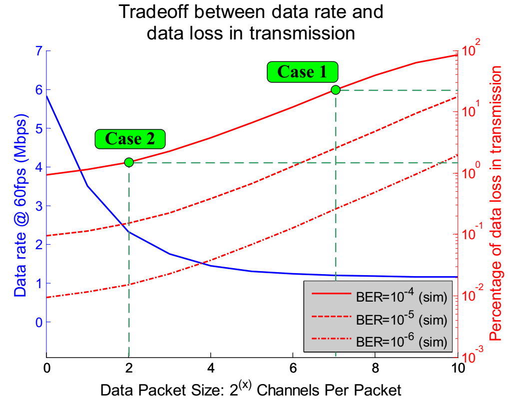
Data transmission error exists in any wireless telemetry system. Because unexpected stimulation caused by corrupted data can lead to serious damage to human tissue, dealing with transmission error is an important issue in the design of any implantable stimulation system. Error detection codes are broadly adopted in implantable systems, such as parity code used in [4], [5] and CRC code in [6]. These schemes pick out the corrupted data packets before they reach the stimulator, but they also inevitably yield data loss. We find that given a wireless channel condition, data loss can be reduced by changing the packet length. The results of our analyses also demonstrate that 5 to 180 times of data loss reduction can be achieved by adding ECC into the data packets.

A. Trade-off between packet length and data loss

The structure of the data packet can be simplified into three parts: header, data and error detection code. Since the fixed-length header and error detection code can be considered as overheads, it seems rational to put in as many data as possible into each data packet in order to increase the data transmission efficiency. However, because the received data packet can only be used when the entire packet passes the error detection check (i.e., not a single bit of error is allowed in the entire packet), the longer the data packet is, the more vulnerable it will be. The two cases in Fig. 2 explain this concept. Case 1 packs data for 128 channels into a single data packet; Case 2 uses 32 smaller packets to carry the same amount of data. Given the lengths of header, data and error detection code are 38 bits, 19 bits and 38 bits (from our protocol, as described in Section IV), respectively, the total number of bits to be transmitted in Case 1 and 2 are 2508 and 4864. The probability of packet fail under a certain bit error ratio (BER) condition in both cases can then be calculated using (1), where L is the length of the data packet in bits.

$$P_{fail} = 1 - (1 - BER)^L \quad (1)$$

When BER=10^{-4} is applied in this equation, 22.2% of the data packet in Case 1 is corrupted; while the data loss in Case 2 is only 1.51%. In other words, sending short packets, although it increases the amount of data, can reduce the data loss by more than 10 times. A complete analysis based on the 1024-channel retinal prosthesis using 3 different BER values is shown in Fig. 3. In this figure, the X-axis represents the data packet length shown in 2^x channels per packet; the Y-axis on the left is the data rate required for sending 60 frames per second (each frame contains data for 1024 channels); the Y-axis on the right is the percentage of data packet failure. It is clearly shown on the figure that sending longer packets, or moving toward the right, conserves the data bandwidth requirement, but also raises the chance of data loss. The two cases illustrated in Fig. 2 are labeled in Fig. 3 with detailed values listed in the table. Therefore, under the same BER condition, data loss can be manipulated by changing the data packet size.



	BER	Packet length (Pixel)	Data rate (Mbps)	Avg. data loss (%)
Case 1	10^{-4}	128	1.20	22.2
Case 2	10^{-4}	4	2.33	1.51

Fig. 3. Shorter data packets are less susceptible to data corruption during wireless transmission. The cost is a bigger portion of overheads and thus higher overall data rate.

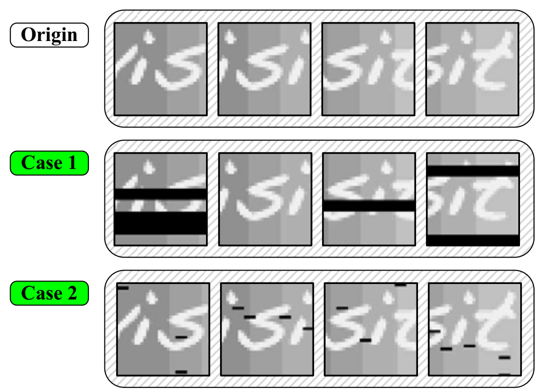


Fig. 4. Simulated 1024-pixel images (32x32) to show the effect of wireless transmission error when different data packet lengths are used. The BER of wireless link is 10^{-4}. (a) original images; (b) data packet length = 128 (channels); (c) data packet length = 4 (channels).

To demonstrate the visual effect of the images with transmission error, we simulate the received image so that it mimics what the patient sees when he/she is panning the head at a constant speed while reading large characters. The image size is 32x32 pixels, which matches the number of channels in our design goal. Wireless transmission error, shown in black, is artificially added based on BER=10^{-4} and two different data packet sizes: 128 and 4. According to our derivation, the average number of data losses in the two cases are 227 and 15.4 channels, respectively. The effects of the error in both cases along with the original image are shown in Fig. 4.

B. Costs and benefits of ECC

Error-correcting code (ECC), although it further increases the overheads of data packets, can reverse certain levels of data corruption during wireless transmission. To understand the costs and benefits of employing ECC, we simulate the data loss under different BER conditions. The ECC we use in the simulation is Hamming code because of its reasonable hardware circuit cost. Since the length of data packet in our design is the integral multiple of 19 bits, we add five bits of Hamming codes into every 19-bit unit of the entire data packet to make 24-bit ECC units. Therefore, any single-bit error that occurs in the 24-bit unit can be corrected. Consequently, there are two conditions that an ECC unit can be precisely recovered: all received bits are correct, or only one bit is corrupted. The equation in (2) sums the probability of both conditions.

$$P_{cor} = (1 - BER)^{24} + (1 - BER)^{23} \times BER \times 24 \quad (2)$$

A data packet can be used only if all the ECC units are correctly recovered, and the probability can be calculated with (3), where the N in the equation refers to the number of ECC units in a data packet. However, because the first 19 bits of each data packet is a special marker (described in Section IV) to identify the starting point of a data packet, the ECC is not inserted into this marker. Therefore, the probability calculated in the simulation is based on the condition that the 19-bit marker is correctly received, and all the ECC units are precisely recovered, as in (4).

$$P_{fail} = 1 - (P_{cor})^N \quad (3)$$

$$P_{fail} = 1 - (1 - BER)^{19} \times (P_{cor})^{(N-1)} \quad (4)$$

The result with and without the ECC code under BER=10^{-4} is shown in Fig. 5. The amount of data is increased by 26.3% (from adding 5-bits of ECC to every 19-bits of data), but the data loss is reduced by at least 5 times. Employing the ECC is especially beneficial to long data packets. In the longest data packet that carries 1024 channels, the probability of packet failure is reduced by 180 times after adding the ECC.

Since adding the ECC greatly reduces the data transmission loss, should we always turn the ECC on? To figure out the answer to this question, we set a target tolerance of data loss, and scan through a wide range of BER values in order to calculate the data rate required to form the data packet. Fig. 6 shows the minimal data rate required to achieve the 5% target tolerance with and without the ECC from BER=10^{-3} to BER=$^{-7}$. We find that data packets with ECCs can easily meet the tolerance requirement in all BER values.

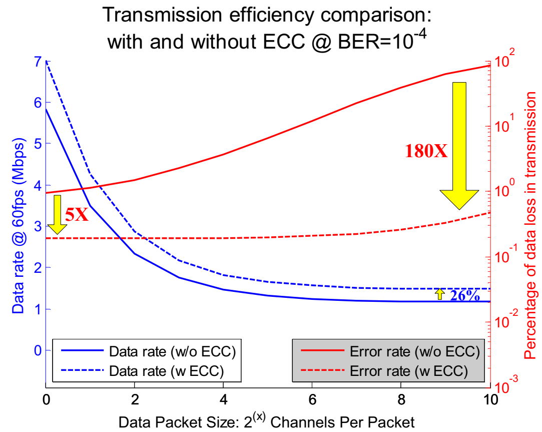


Fig. 5. Data rate and data loss probability comparison with and without ECC. The data rate is increased by 26% with ECC, while the data loss is reduced by 5 to 180 times, depending on the packet size.

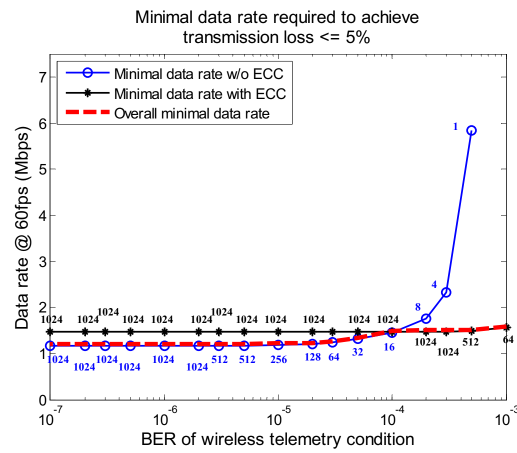


Fig. 6. The minimal data rate required to transmit stimulation parameters for 1024 channels at 60Hz refresh rate with 5% data loss tolerance. The number labeled beside each dot is the data packet length used to achieve the target tolerance.

If the ECC is not used, the 5% target cannot be met at BER=10^{-3}, and the packet length needs to be shortened in order to reduce the error rate to 5% or less for BER= 5×10^{-3} to BER= 2×10^{-3}. This greatly increases the overheads in the data packet and thus increases the overall data rate However, if the wireless channel condition is better, i.e. BER $\leq 10^{-4}$, long data packets can achieve the required target tolerance even without the ECC, so the extra bits contributed by the ECC become a burden to the overall data rate. Therefore, for a 5% target tolerance, the ECC is preferable when BER $> 10^{-4}$, and less useful if BER $\leq 10^{-4}$. This crossover point shifts left if the target tolerance is more stringent, and shifts right when it's loose. This crossover point provides a good reference for whether the ECC should be turned on.

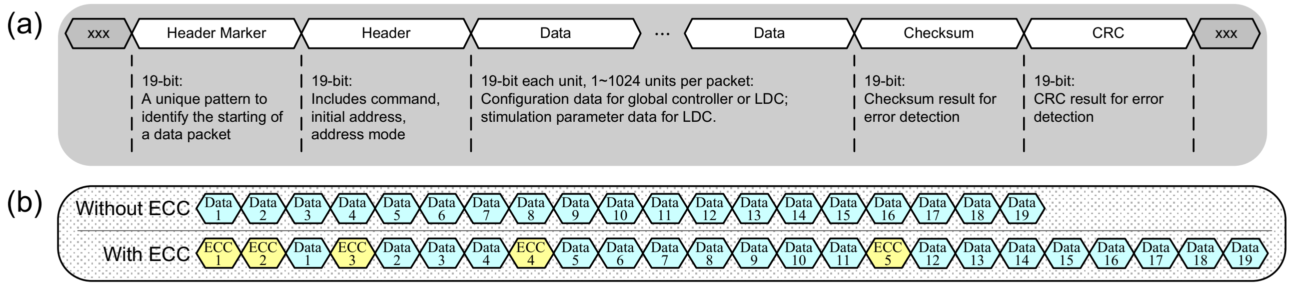


Fig. 7. (a) The data packet protocol used in the retinal prosthesis for wireless data transmission. The length of each unit is 19 bits; (b) Optional ECC (5 bits) can be added into each unit to reduce data loss.

III. Data Transmission Protocol

The detailed data packet protocol is shown in Fig. 7(a). Each data packet is composed of five parts: header marker, header, data, checksum, and cyclic redundancy check (CRC). A header marker is used for the receiver to identify the beginning of a data packet; a header carries the command, initial address and address mode; the checksum and the CRC are used to confirm that the data packet is not corrupted during transmission. These fixed-length overheads are required for each data packet, long or short.

For each channel in the stimulator array, 19-bit-long data are required to define all stimulation parameters of a biphasic current pulse, including pulse width, pulse amplitude, pulse polarity, and inter-phase delay. While some of the parameters are converted from the brightness information received by the camera, others can be programmed by the researchers or physicians.

According to our analysis shown in the previous section, the packet length plays a critical role in the probability of data loss, as the BER condition usually changes from time to time, we decided to make the length of each data packet variable, which increases the design complexity of the receiver circuit. Each packet can carry data for 1 to 1024 channels at a time, and this feature gives us the flexibility to adaptively alter the packet size based on the BER conditions.

An optional 5-bit Hamming code can be added into the data packet for the purpose of error correction, as shown in Fig. 7(b). This applies to every 19-bit unit in the packet except the header marker. By adding the ECC, a single-bit error happening in each unit can be correctly recovered. Due to the limitation of the Hamming code, error correction cannot be properly performed if more than one bit of error occurs in one unit. Therefore, an overall error detection scheme–checksum and CRC in our design–is still required to check the data packet after each ECC unit is recovered.

IV. Data Packet Receiver Circuit Implementation

After we define the data packet protocol, a corresponding circuit that can receive and decode the data packets is also implemented using Verilog hardware description language. The first stage of the circuit is a header marker detector. When the packet start is detected, the rest of the data packet optionally flows through an error correction circuit, which recovers each data unit and removes the 5-bit ECC. The result is sent to a digital controller (detailed described in [2]) for command decoding and data dispatch. An error detection circuit runs throughout the entire packet to make sure the recovered data are correct. This circuit is implemented in 0.18μm CMOS semiconductor technology, and the area for the core circuit with the entire digital controller is 330μm $\times$ 262μm.

V. Conclusions and Future Works

To improve the quality of the artificially induced visual perception for a 1024-channel retinal prosthesis, we analyzed the trade-off between data loss and packet length. The results show that data loss can be reduced by more than 10 times by sending short data packets rather than long ones. With a given target data loss, we can also optimize the packet configuration (size and ECC on/off) to achieve lowest data rate based on the wireless channel conditions. The circuit design and simulation for the data protocol receiver is done, and we expect to incorporate this circuit into our next generation retinal IC for fabrication in the near future.

VI. Acknowledgments

The authors gratefully acknowledge the chip fabrication and packing services provided by TSMC and the digital standard cells library supported by ARM.

References

[1] J. D. Weiland and M. S. Humayun, "Visual prosthesis," *Proc. of the IEEE*, vol. 96, no. 7, pp. 1076–1084, 2008.

[2] K. Chen and W. Liu, "Highly programmable digital controller for high-density epi-retinal prosthesis," *Proc. of the IEEE Engineering in Medicine and Biology Conf.*, pp. 1592–1595, Sep. 2009.

[3] K. Chen, Z. Yang, L. Hoang, J. D. Weiland, M. S. Humayun, and W. Liu, "An integrated 256-channel epiretinal prosthesis," *IEEE J. of Solid-State Circuits*, vol. 45, no. 9, pp. 1946–1956, 2010.

[4] M. Ghovanloo and K. Najafi, "A modular 32-site wireless neural stimulation microsystem," *IEEE J. of Solid-State Circuits*, vol. 39, no. 12, pp. 2457–2466, 2004.

[5] K. D. Wise, A. M. Sodagar, Y. Yao, M. N. Gulari, G. E. Perlin, and K. Najafi, "Microelectrodes, microelectronics, and implantable neural microsystems," *Proc. of the IEEE*, vol. 96, no. 7, pp. 1184–1202, 2008.

[6] M. Ortmanns, A. Rocke, M. Gehrke, and H.-J. Tiedtke, "A 232-channel epiretinal stimulator asic," *IEEE J. of Solid-State Circuits*, vol. 42, no. 12, pp. 2946–1959, 2007.